



# Administering Jira Data Center and Server 8.20 applications

# Contents

Administering Jira Data Center and Server 8.20 applications .....	8
Jira applications and project types overview .....	9
Getting started as an administrator .....	11
Setting up your instance .....	12
Creating a project .....	14
Adding new users .....	15
Managing permissions .....	16
Installing Jira applications .....	18
Jira applications installation requirements .....	20
Installing Java .....	22
Supported platforms .....	24
End of support announcements .....	29
Install a Jira trial .....	39
Installing Jira applications on Windows .....	41
Uninstalling Jira applications from Windows .....	45
Installing Jira applications on Windows from Zip File .....	46
Installing Jira applications on Linux .....	52
Uninstalling Jira applications from Linux .....	56
Installing Jira applications on Linux from Archive File .....	57
Unattended installation .....	64
Installing additional applications and version updates .....	66
Troubleshooting installation .....	68
Connecting Jira applications to a database .....	69
Connecting Jira Data Center to Amazon Aurora .....	70
Connecting Jira applications to Azure SQL .....	74
Connecting Jira applications to PostgreSQL .....	78
Connecting Jira applications to MySQL 5.7 .....	83
Connecting Jira applications to MySQL 8.0 .....	89
Connecting Jira applications to Oracle .....	95
Connecting Jira applications to SQL Server 2016 .....	98
Connecting Jira applications to SQL Server 2017 .....	102
Tuning database connections .....	106
Surviving connection closures .....	114
Encrypting database password .....	116
Basic encryption .....	117
Advanced encryption .....	119
Creating your own Cipher .....	123
Switching databases .....	126
Connecting Jira applications to SQL Server 2019 .....	128
Installing Jira Data Center .....	132
Migrate from Server to Data Center .....	137
Running the setup wizard .....	139
Licensing and application access .....	143
License compatibility .....	145
Extending Jira applications .....	147
Integrating with development tools .....	149
Integrating with development tools using DVCS .....	152
Integrating with development tools using app links .....	173
Integrating with other tools .....	175
Listeners .....	177
Managing webhooks .....	179
Services .....	180
Using AppLinks to link to other applications .....	182
Administering projects and links across multiple applications .....	185
Integrating with collaboration tools .....	190
Managing apps .....	194
Upgrading Jira applications .....	195
Reasons to upgrade .....	197

Upgrade checklist	200
Upgrade matrix	207
Upgrade methods	228
Creating a test environment for Jira	231
Preparing for the upgrade	236
Upgrading Jira (installer)	239
Upgrading Jira (manual)	245
Upgrading Jira Data Center (installer)	250
Upgrading Jira Data Center (manual)	257
Upgrading Jira Data Center with zero downtime	263
Zero downtime upgrade checklist	266
Upgrade task troubleshooting	267
Zero downtime upgrade FAQs	268
Upgrading Jira with a fallback method	271
Rolling back a Jira application upgrade	274
Establishing staging server environments for Jira applications	275
Migrating Jira applications to another server	276
Migrating from Jira Cloud to Server applications	280
Migrating from Jira Server to Cloud	283
Running Jira Data Center on a Kubernetes cluster	284
Getting started with Jira Data Center on AWS	286
Administering Jira Data Center on AWS	292
Upgrading Jira Data Center on AWS	294
Getting started with Jira Data Center on Azure	297
Administering Jira Software Data Center on Azure	302
Federating Jira - managing multiple instances	305
Layout and design	312
Configuring the look and feel of your Jira applications	313
Configuring an announcement banner	315
Configuring the default dashboard	316
Using dashboard gadgets	317
Adding a gadget to the directory	320
Subscribing to another application's gadgets	323
Choosing a default language	325
Translating Jira	327
Configuring the default issue navigator	328
Creating links in the application navigator	330
Configuring the user default settings	331
User management	332
Managing users	333
Create, edit, or remove a user	334
Assign users to groups, project roles, and applications	339
Monitor a user's activity	341
Manage password security	342
Prevent automatic login	344
SAML SSO for Jira Data Center applications	346
Anonymizing users	349
Retrying anonymization	355
Enabling public signup and CAPTCHA	357
Managing groups	360
View, create, or delete a group	361
Modify group membership	362
Assign group access to a project role	363
Manage group access to applications	364
Advanced user management	366
Allowing connections to Jira for user management	367
Diagrams of possible configurations for user management	370
Managing nested groups	377
User management limitations and recommendations	380
Configuring user directories	384
Configuring the internal directory	387
Connecting to an LDAP directory	389
Configuring an SSL connection to Active Directory	398
Reducing the number of users synchronized from LDAP to JIRA applications	409

Connecting to an internal directory with LDAP authentication	411
Connecting to Crowd or another Jira application for user management	417
Managing multiple directories	425
Migrating users between user directories	427
Synchronizing data from external directories	430
Configuring projects	432
Defining a project	433
Editing a project key	438
Changing the project key format	440
Configuring issues	442
Configuring built-in fields	445
Defining issue type field values	446
Defining priority field values	451
Defining resolution field values	457
Defining status field values	458
Translating resolutions, priorities, statuses, and issue types	459
Issue fields and statuses	460
Configuring issue-level security	464
Configuring permissions	467
Managing global permissions	470
Managing project permissions	474
Customizing Jira Service Management permissions	480
Resolving Jira Service Management permission errors	482
Using Manage Sprints permission for advanced cases	485
Sprint permissions and defined processes	487
Managing project roles	491
Managing project role membership	494
Allowing anonymous access to your project	495
Managing components	496
Archiving a component	499
Managing versions	501
Creating release notes	504
Project screens, schemes and fields	505
Managing custom fields	507
Adding custom fields	509
Configuring custom field contexts	512
Editing or deleting custom fields	514
Translating custom fields	516
Analyzing the usage of custom fields	517
Optimizing custom fields	520
Managing system fields	524
Configuring contexts and default values for the Description field	526
Viewing and configuring screens	528
Specifying field behavior	529
Associating field behavior with issue types	533
Configuring renderers	536
Defining a screen	542
Associating a screen with an issue operation	545
Associating screen and issue operation mappings with an issue type	548
Creating a notification scheme	551
Using the issue collector	555
Advanced use of the Jira issue collector	559
Working with workflows	569
Managing your workflows	573
Configuring workflow schemes	576
Sharing your workflow	579
Advanced workflow configuration	582
Working in text mode	590
Adding a custom event	593
Configuring the initial status	596
Configuring workflow triggers	597
Using validators with custom fields	608
Using XML to create a workflow	609
Workflow properties	610

Configuring Jira Service Management approvals .....	612
Archiving an issue .....	617
Archiving a project .....	623
Importing and exporting data .....	626
Migrating from other issue trackers .....	627
Importing data from CSV .....	628
Commonly asked CSV questions and known issues .....	638
How to import CSV data with PVCS command .....	639
Importing data from Excel .....	640
Importing data from Bitbucket .....	641
Importing data from Github .....	642
Importing Data from Asana .....	643
Importing data from TFS or Visual Studio .....	644
Importing data from Rally .....	645
Importing data from VersionOne .....	646
Importing data from YouTrack .....	647
Importing data from Axosoft .....	648
Importing data from Pivotal Tracker .....	649
Importing data from Bugzilla .....	650
Importing data from FogBugz On Demand .....	651
Importing data from FogBugz for your Server .....	652
Importing data from Trac .....	653
Importing data from Redmine .....	654
Importing data from BaseCamp .....	655
Importing data from JSON .....	656
Importing data from Mantis .....	664
Moving or archiving individual projects .....	665
Splitting Jira applications .....	666
Exporting issues from Cloud to Server .....	667
Exporting issues from Server to Cloud .....	669
Migrating data with 3rd party apps .....	670
Migrating data with Adaptavist .....	671
Promoting configuration changes from staging to production .....	672
Migrating projects to another Jira instance .....	683
Merging Jira instances .....	697
Migrating data with Botron .....	699
Promoting Jira configuration from development to production .....	700
Migrating Jira projects .....	710
Consolidating multiple Jira instances .....	719
Configuring Jira application emails .....	720
Configuring email notifications .....	721
Configuring an SMTP mail server to send notifications .....	726
Customizing email content .....	731
Templates: Batched issue notifications and other events .....	734
Templates: Separate issue notifications and other events .....	740
Examples: Customizing email content .....	748
Customizing email content in Jira Server .....	750
Adding custom fields to emails .....	753
Different types of email notifications in Jira .....	757
Creating issues and comments from email .....	759
Configuring Jira applications to receive email from a POP or IMAP mail server .....	774
Integrating with OAuth 2.0 .....	778
Jira system administration .....	781
System administration .....	782
Finding your Server ID .....	783
Increasing Jira application memory .....	784
Using the database integrity checker .....	791
Precompiling JSP pages .....	793
Logging and profiling .....	794
Logging email protocol details .....	797
Backing up data .....	799
Automating Jira application backups .....	801
Preventing users from accessing Jira applications during backups .....	803
Restoring data .....	804

Restoring a project from backup	805
Anonymising Jira application data	814
Restoring data from an xml backup	816
Restoring information from a native backup	818
Search indexing	819
Re-indexing after major configuration changes	823
Using robots.txt to hide from search engines	824
Control anonymous user access	825
Licensing your Jira applications	828
Viewing your system information	830
Live monitoring using the JMX interface	836
Monitoring database connection usage	842
Monitor your instance with Jira diagnostics plugin	845
Viewing Jira application instrumentation statistics	848
Generating a thread dump	853
Finding your Jira application Support Entitlement Number (SEN)	859
Auditing in Jira	860
Audit log events in Jira	866
Audit log integrations in Jira	877
Data pipeline	880
Data pipeline export schema	887
Make the most of the data pipeline with the DevOps dashboard	899
Deploy the DevOps dashboard in Tableau	907
Deploy the DevOps dashboard in PowerBI	912
Important directories and files	915
Jira application installation directory	919
Jira application home directory	921
Setting your Jira application home directory	923
Integrating Jira applications with a Web server	925
Integrating Jira applications with IIS	926
Integrating Jira with Apache	931
Securing Jira applications with Apache HTTP Server	949
Using Apache to limit access to the Jira administration interface	950
Using Fail2Ban to limit login attempts	961
Changing Jira application TCP ports	963
Connecting to SSL services	965
Running Jira applications over SSL or HTTPS	974
Configuring security in the external environment	987
Data collection policy	988
Jira Admin Helper	989
Raising support requests as an administrator	992
Start and Stop Jira applications	996
Managing LexoRank	999
Jira cluster monitoring	1003
Scheduler administration	1009
Configuring global settings	1011
Configuring time tracking	1012
Configuring Jira application options	1015
Configuring advanced settings	1020
Configuring the Base URL	1023
Configuring the administrator contact form	1024
Setting properties and options on startup	1026
Recognized system properties for Jira applications	1035
Advanced Jira application configuration	1040
Changing the constraints on historical time parameters in gadgets	1042
Changing the default order for comments from ascending to descending	1043
Limiting the number of issues returned from a search view such as an RSS feed	1044
Configuring file attachments	1046
Configuring issue linking	1051
Configuring issue cloning	1054
Configuring the allowlist	1055
Configuring sub-tasks	1057
Managing filters	1060
Managing dashboards	1063

Enabling logout confirmation .....	1065
Rich text editing .....	1066
Configuring terminology .....	1069
Server optimization .....	1071
Configuring secure administrator sessions .....	1072
Improving instance stability with rate limiting .....	1073
Adjusting your code for rate limiting .....	1080
Jira application cookies .....	1083
Preventing security attacks .....	1085
Using the Jira application configuration tool .....	1087
Running Jira applications as a Windows service .....	1092
Tuning garbage collection (GC) .....	1098
Jira Data Center documentation .....	1099
Running Jira Data Center on a single node .....	1100
Moving back to Server .....	1103
Running Jira Data Center in a cluster .....	1105
Set up a Jira Data Center cluster .....	1111
Adding and removing Data Center nodes .....	1115
Performance and scaling .....	1117
Best practices for scaling Jira Software .....	1118
Performance and scale testing .....	1121
Use a CDN with Atlassian Data Center applications .....	1130
Configure your CDN for Jira Data Center .....	1134
Jira Data Center monitoring .....	1138
Security overview and advisories .....	1146
Getting help .....	1148

# Administering Jira Data Center and Server 8.20 applications

A one-stop shop for administering all of your Jira applications: Jira Core, JiraSoftware, and JiraService Management.

---

## Get started

New to Jira? Check out our guide for new administrators.

[View guide](#)

## Whats new

Time to upgrade? Get the lowdown on the latest and greatest in Jira8.20.

[View latest changes](#)

# Jira applications and project types overview

The Jira family of applications are built to deliver a tailored experience to their user. Jira Core is the default application of Jira, and will always be present in a Jira instance. You may also choose to include other applications in your instance, such as Jira Software or Jira Service Management. A user may require access to one, all, or any combination of these applications.

Note that as Jira Core is the default application, if you have a license for Jira Software or Jira Service Management, your users automatically have access to Jira Core *without* requiring an additional license. For example, a Jira Software user can view development information on an agile board, and can also view business projects.

## Application features and project types

Each application delivers a tailored experience for its users, and has an associated project type, which in turn, offers application-specific features. Below is a list of the project types, and their associated application-specific features.

Application	Project type	Application-specific feature set
Jira Core	 Business projects	<ul style="list-style-type: none"><li>• Available to all licensed users of Jira</li></ul>
Jira Software	 Software projects	<ul style="list-style-type: none"><li>• Integration with development tools</li><li>• Agile boards</li><li>• Release hub for software version release</li></ul>
Jira Service Management	 Service projects	<ul style="list-style-type: none"><li>• Service Level Agreements (SLAs)</li><li>• A customizable web portal for customers</li><li>• Permission schemes allowing customer access</li></ul>

## Application features and users

All users that can log in to a Jira instance will be able to see all the projects in that instance (pending permissions), but they will only be able to see the application-specific features when they have application access. For example, a Software project is able to display information from linked development applications, such as Bitbucket and FishEye on a Software project, and you can create agile boards, but this information is only viewable by a Jira Software user. A Jira Core user would be able to see the Software project, but would not be able to see the application-specific features, like agile boards or development information. Likewise, a Jira Software user would not be able to see any Jira Service Management application-specific features on a service project only a basic view of the project and its issues.



- Only a Jira administrator can create a project for an installed application. They do not need application access to create the project, but they do need application access if they'd like to view or use the project.
- Anonymous users will have access equivalent to Jira Core users. In other words, they can view issues and work in any type of project, but they won't see application-specific features, e.g. agile boards, which are Jira Software-specific features. To know how to allow anonymous users access to projects, see [Allowing anonymous access to your instance](#).

A list of the applications, their default user groups, and their project's application-specific features is listed below:

			Jira Core	Jira Software	Jira Service Management
			jira-core-user	jira-software-user	jira-servicedesk-agent
<b>Business Projects</b> 	Project level	View	✓	✓	✓
	Issue level	Create	✓	✓	✓
		View	✓	✓	✓
		Comment	✓	✓	✓
		Transition	✓	✓	✓
Jira Gadgets	View	✓	✓	✓	
<b>Software Projects</b> 	Project level	View	✓	✓	✓
	Issue level	Create	✓	✓	✓
		View	✓	✓	✓
		Comment	✓	✓	✓
		Transition	✓	✓	✓
		View development information	✓	✓	✓
		View release information	✗	✓	✗
	Board level	Create	✗	✓	✗
		View	✗	✓	✗
	Jira Software gadgets	View	✗	✓	✗
<b>Service Projects</b> 	Project level	View	✓	✓	✓
	Issue level	Create	✓	✓	✓
		View	✓	✓	✓
		Comment	✓	✓	✓
		Transition	✗	✗	✓
	SLA level	Create	✗	✗	✓
		View	✗	✗	✓
	Queue level	Create	✗	✗	✓
		View	✗	✗	✓
	Jira Service Management gadgets	View	✗	✗	✓

# Getting started as an administrator

This tutorial will get you up to speed with Jira, regardless of whether you're using Jira Core, Jira Software, or Jira Service Desk. All administrators should complete it to understand the basic concepts of managing a Jira instance.

**Audience**

Administrators

**Time**

45 minutes

Here's what you can count on learning:

1. How to set up an evaluation instance of Jira
2. How to add new users
3. How to create a project and customize it
4. How to manage permissions

By the end of this tutorial, you'll have a fully functioning Jira instance, several users with different access permissions, and you'll have created your first project.

---

**Ready to dive in and get your hands dirty? Get started and learn how to set up your own Jira.**

[Let's get started](#)

# Setting up your instance

1. Setting up your instance
2. Creating a project
3. Adding new users
4. Managing permissions

As a first step, you'll set up an evaluation instance of Jira. We'll guide you through three simple steps to get Jira up and running in no time!



## On this page:

- [Before you begin](#)
- [Download the installer](#)
- [Install your Jira application](#)
- [Set up your Jira application](#)

If you're ready to set up a production Jirasite or you want more control, check out our [full installation guides](#).

## Before you begin

Our installers come with all the bits and pieces you need to run the application, but there's a few things you'll need to get up and running:

- A computer or laptop with a supported operating system - you'll be installing Jira so you'll need admin rights.

You can install Jira on a Windows or Linux operating system.

Apple Mac isn't supported for production sites, but if you're comfortable setting up applications on your Mac from scratch, you can download the `tar.gz` file and follow the instructions for [Installing Jira applications on Linux from Archive File](#) as the process is similar.

- A supported web browser - you'll need this to access Jira, we support the latest versions of Chrome and Mozilla Firefox, Internet Explorer 11, and Microsoft Edge.
- A valid email address - you'll need this to generate your evaluation license and create an account.

Ready to get going? Let's start with grabbing the installer.

## Download the installer

Head to [www.atlassian.com/software/jira/core/download](http://www.atlassian.com/software/jira/core/download) and download the installer for your operating system.

## Install your Jira application

The installer allows you to choose Express or Custom installations.

The Custom installation allows you to pick some specific options for Jira, but for this guide we'll use the Express installation.

1. Run the installer - we recommend running with a Windows administrator account. If prompted, make sure you allow the installer to make changes to your computer. This will allow you to install Jira as a service.
2. Choose **Express Install**, then click **Next**.
3. Once installation is complete, it will ask you if you want to open Jira in your browser. Make sure this option is selected then click **Done**.
4. Jira will open in your default browser, and you're ready to start the set up wizard.

1. Change to the directory where you downloaded Jira then execute this command to make it executable:

```
$ chmod a+x atlassian-jira-software-X.X.X-x64.bin
```

Where `jira-software.X.X.X` is the JIRA version you downloaded.

2. Run the installer - we recommend using `sudo` to run the installer as this will create a dedicated account to run Jira and allow you to run Jira as a service.

```
$ sudo ./atlassian-jira-software-X.X.X-x64.bin
```

3. When prompted, choose **Express Install** (option 1).
4. Once installation is complete head to <http://localhost:8080> in your browser to begin the setup process.

## Set up your Jira application

The set up wizard is the last step in getting Jira up and running. You'll need your email address to generate your evaluation license.

1. Select **Set it up for me** and then **Continue to My Atlassian**. This will allow Jira to set up everything it needs to run, including an H2 database.
2. Create an account (or log in if you already have an Atlassian ID account).
3. Follow the prompts to **generate a license** for the JIRA application you want to try, and apply it to your new installation.
4. Enter and confirm the details you want to use for your administrator account, and click **Next**.
5. It will take a few minutes to get everything connected and operational.

**That's it! Let's now create your first project.**

[Next](#)

# Creating a project

1. [Setting up your instance](#)
2. Creating a project
3. Adding new users
4. Managing permissions

A Jira project is a container that holds issues. Issues can be viewed as the packets of work required within a project. To create issues, you must have an available project to contain them. Jira comes with several default project types with preconfigured workflows and issue types, so you can quickly get your project up and running. In this step of the tutorial, you will use the project management template to help your team plan, organize, and collaborate on their work.

Note that creating and configuring a project is done by an administrator. A project administrator controls user access to the project, and can only configure certain aspects of the look and feel of the project. You should still be logged in to Jira as an administrator from the previous step. If not, log into your administrator account.

## Create a project

When creating a project, you will need to give it a name, a key, and a project lead. The title can be as descriptive as you want, and the key should be something meaningful. The project lead is usually the project manager, but can effectively be any user you select when creating the project.

1. If you're following from the previous step, you should see different project options on the welcome screen. Click **Create new project**.
2. Select **Project management** as the project type.
3. Enter **Dragon Design Tees** as the project name. Note that Jira creates a Project key for you, but you can overwrite this if you want to.
4. Select **Submit** to create your new project.

### About project keys

Each project has a unique *name* (e.g. **Dragon Design Tees**) and a unique *key* (e.g. **DDT**). The project key becomes the first part of that project's *issue keys*, e.g. **DDT-1**, **DDT-2**, etc.

## Customize your project

In this step, you will be customizing your project avatar and project details to help your team identify the project more easily. These customizations are helpful if you have several projects in your Jira instance. If you have navigated away from your project, simply go to **Projects > Dragon Design Tees**.

1. Click **Administration**  **> Projects**.
2. Select **Edit** next to your project.
3. Click the **Avatar** image.
4. Select an available icon or upload an image.
5. Enter a URL and Description for your project to make it easier for your team to identify. Note that these fields are optional and only for display.
6. Click **Save details** to save your changes.

---

**Congratulations! You've now created and customized your first project. Next, we'll add users to your project and look at how you can set up and restrict access to projects.**

[Next](#)

# Adding new users

1. [Setting up your instance](#)
2. [Creating a project](#)
3. Adding new users
4. Managing permissions

Working alone isn't much fun, so let's add some test users to your Jirasite. You can add users directly, or allow new users to sign up themselves. In this step in the tutorial, you'll add three users directly to your site.

## Add a few users

You will be adding three users: **Jason**, **Kate** and **Emma**. You can add more or choose your own usernames if you like, but please note that we will be referring to these usernames later in the tutorial. You can always disable or delete any users you set up.

If you've logged out of Jira, log in with the administrator account you created.

1. Navigate to the **User Management** screen by selecting **Administration** (⚙️) > **User management**.
2. Choose **Create User** to add a new user. Specify the username as **jason**. Set the rest of the fields to whatever you want. You're going to be creating a couple more users, so check the **Create another** checkbox before selecting **Create user**.
3. Now create two more users, with the usernames **emma** and **kate**, following the same process outlined above.

You should have a screen that looks something like this:

### Users

Invite users Create user

Filter users:  In group:  Application access:  Status:  Users per page:  Filter Reset

Displaying users 1 to 4 of 4.

Full name	Username	Login details	Group name	Applications	Directory	Actions
Emma	emma emma@atlassian.com	Never logged in	jira-core-users	JIRA Core	JIRA Internal Directory	Edit ...
Jason	jason jason@atlassian.com	Never logged in	jira-core-users	JIRA Core	JIRA Internal Directory	Edit ...
Kate	kate kate@atlassian.com	Never logged in	jira-core-users	JIRA Core	JIRA Internal Directory	Edit ...
Tomek	admin tomek@atlassian.com	Count: 25 Last: 3 minutes ago	jira-administrators jira-core-users jira-servicedesk-users jira-software-users	JIRA Service Desk JIRA Software	JIRA Internal Directory	Edit ...

**i** Note that usernames are **not** case sensitive. Emma can enter her username as Emma, emma, or even EmMA to log into Jira. Passwords, on the other hand, are case sensitive.

**Well done! You've added three new users to your Jira instance. Next, you'll learn how to manage access to your project with site and project permissions.**

[Next](#)

# Managing permissions

1. [Setting up your instance](#)
2. [Creating a project](#)
3. [Adding new users](#)
4. Managing permissions

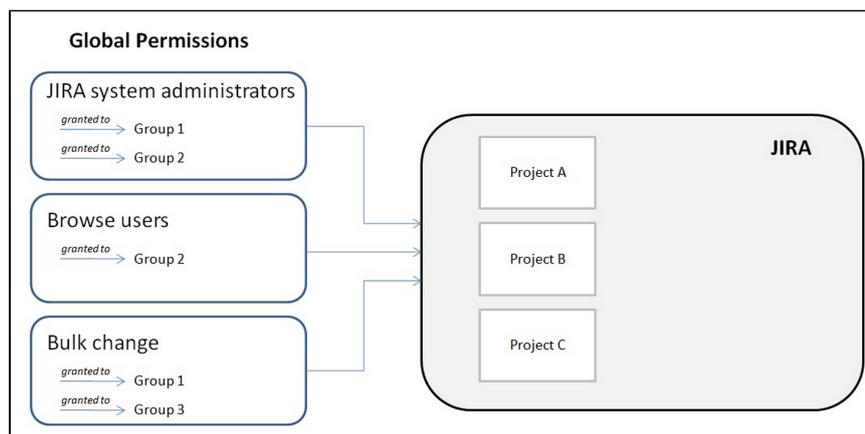
You won't want every user in your team to have the same level of access to Jira. For example, you may want to restrict who can administer Jira, or prevent users from viewing a project. In this step, you will learn about the different permissions in Jira and set permissions for a new project.

## Overview of roles, groups, and users

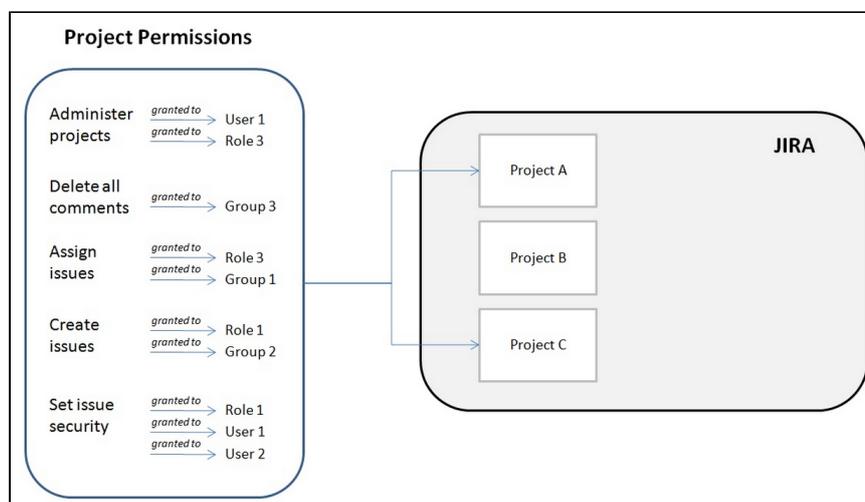
A role is a project-specific set of groups and/or individual users. In our example of the design project in the t-shirt business, all product managers need to be able to assign work (issues) across all projects, while senior designers need to be able to assign work on specific design projects. In Jira, you can define a product manager role that includes all product managers. You can then define a set of permissions with the 'Assign issue' permission for this role, and apply this set of permissions to all projects. Individual senior designers can be added to the product manager role on each project, as needed.

## Overview of global and project permissions

Global permissions cover a small set of functions that affect all projects in Jira (for example, permission to administer Jira). They can only be assigned to groups:



Project permissions cover a set of more granular functions that affect a single project in Jira. For example, permission to create issues in a project. They can be assigned to groups, users and project-specific roles:



Now let's put this into practice! You're going to go through the tasks involved to use project permissions to hide a new, secret t-shirt design project from some of your users.

## Create a new project role

This project role will only contain users that you want to view a particular project. We will assign permissions to this role in the next step.

1. Select **Administration** (⚙️) > **System**.
2. In the security section, select **Project roles**.
3. Below the existing project roles, add another project role named "Review". Leave the Description field blank for now and select **Add project role**.
4. Select **Manage Default Members** and then, under Default Users, select **Edit** to add yourself and **Jason** to the Review project role. Do not add Kate or Emma.

## Configure a new permission scheme

The 'Browse Projects' permission controls whether a user can browse a project, i.e. whether they can view the project. Let's assign this permission to your new project role.

1. Select **Administration** (⚙️) > **Issues** > **Permission Schemes**.
2. Copy the **Default Permission Scheme**.
3. Edit the copied permission scheme and change the name to **Confidential Permission Scheme**. Select **Update**.
4. Select **Permissions** for the **Confidential Permission Scheme**. For the **Browse Projects** permission:
  - Click **Remove** for 'Application Role (Any logged in user)'.
  - Click **Edit**, select **Project Role**, and choose **Review** in the drop-down. Click **Grant**.

## Associate the scheme with a new project

For the last step, let's associate the permission scheme with your new project.

1. Select **Projects** > **Create Project** and choose **Task management**.
2. Name the project **Top Secret Tee** and **Submit**.
3. In the bottom-left corner, select **Project settings** > **Permissions**.
4. On the Default Permission Scheme screen, select **Actions** > **Use a different scheme**.
5. Set the Scheme to **Confidential Permission Scheme**, and click **Associate**.

The only users that will be able to browse your new project are Jason and yourself. Note that default members are only added to a role for new projects. You can also use this approach to restrict users from creating issues, adding comments, closing issues, etc, in a project.

---

**Well done! You've completed the Administrator Getting started tutorial!**

# Installing Jira applications

Before you start

Before installing, please review the [supported platforms](#) and [Jira applications installation requirements](#) pages.

Choose your install method

Install method	Is this right for you?
<b>Install a Jira trial</b> <ul style="list-style-type: none"><li>• <a href="#">Windows</a>, <a href="#">Linux</a> or <a href="#">OS X</a></li></ul>	This is the fastest way to get Jira up and running. If you're evaluating Jira, use this option. You don't need an external database to install a Jira trial.  You can also <a href="#">try Jira on the Cloud for free</a> .
<b>Install Jira using an installer</b> <ul style="list-style-type: none"><li>• <a href="#">Windows</a></li><li>• <a href="#">Linux</a></li></ul>	This option uses an installer, and is the most straightforward way to get your production site up and running on a Windows or Linux server.
<b>Install Jira from a zip or archive file</b> <ul style="list-style-type: none"><li>• <a href="#">Windows</a></li><li>• <a href="#">Linux</a></li></ul>	This option requires you to manually install files and configure some system properties. It gives you the most control over the install process. Use this option if there isn't an installer for your operating system.
<b>Unattended Jira installation</b> <ul style="list-style-type: none"><li>• <a href="#">Windows</a> or <a href="#">Linux</a></li></ul>	You can use a configuration file from an existing Jira installation to create an identical installation with no user input. This is useful for re-installing Jira on production with the same configuration as an existing test installation.
<b>Run Jira in a Docker container</b> <ul style="list-style-type: none"><li>• <a href="#">Jira Core: Docker</a></li><li>• <a href="#">Jira Software : Docker</a></li><li>• <a href="#">Jira Service Management: Docker</a></li></ul>	This option gets Jira up and running in no time using a pre-configured Docker image. Head to <a href="https://docs.docker.com/">https://docs.docker.com/</a> to find out more about Docker.  Atlassian supports running Jira in a Docker container, but we cannot offer support for problems which are related to the environment itself.

<p><b>Install Jira Data Center on a single node</b></p> <p>It's the same as a Server installation:</p> <ul style="list-style-type: none"> <li>▪ Installer: <a href="#">Windows / Linux</a></li> <li>▪ Archive file: <a href="#">Windows / Linux</a></li> </ul>	<p>You don't need high availability or disaster recovery, but could use features that are exclusive to Data Center. Deploying Data Center on a single node is the same as installing Jira Server, just with a different license, which unlocks new features.</p> <p>Learn more at <a href="#">Running Jira Data Center on a single node</a>.</p>
<p><b>Install Jira Data Center in a cluster</b></p> <ul style="list-style-type: none"> <li>• <a href="#">Kubernetes</a></li> <li>• <a href="#">Windows and Linux</a></li> <li>• <a href="#">AWS</a></li> <li>• <a href="#">Azure</a></li> </ul>	<p>Jira Data Center with a cluster of nodes is designed for enterprises with large or mission-critical deployments that require continuous uptime, instant scalability, and performance under high load. It can be hosted on your own infrastructure or deployed to Kubernetes, AWS or Azure.</p> <p>Learn more at <a href="#">Running Jira Data Center in a cluster</a>.</p>

**Note:** We do not support installing Jira as a production system on macOS. However, if you want to set up Jira on a macOS for evaluation purposes, follow the instructions for [Installing Jira applications on Linux from Archive File](#). There are no limitations to using Jira on a Mac with any one of the [supported browsers](#).

### Installing an additional Jira application

Once you have installed your initial Jira application, it's possible to [install additional applications](#) directly through the Versions and licenses page.

# Jira applications installation requirements

- [Jira applications installation requirements](#)
- [Client-side requirements](#)
- [Server-side requirements for evaluation purposes](#)
- [Server-side requirements for production](#)
- [Next Steps](#)

No hardware? No problem! Try using Jira applications in the Cloud.

- No installation required, get started in 5 minutes
- Option to migrate to your own server later
- Choose from a set of supported apps to install

 [Start My Free Trial](#)

## Jira applications installation requirements

Jira is a 'web application', meaning it runs centrally on a server, and users interact with it through web browsers from any computer on the same network. As such, Jira must be able to communicate and authenticate with itself. If you're upgrading to Jira 8.20 be sure to review the latest release and upgrade notes [here](#).

Please read the [Supported platforms](#) page for Jira applications, which lists the required server and client software supported by Jira applications for:

- Browsers (client-side)
- Java platforms (JDK/JRE) (server-side)
- Operating systems (server-side)
- Application servers (server-side)
- Databases (server-side)

Please also read the information below regarding server and client software and hardware requirements for Jira.

### Client-side requirements

<b>Browser</b>	Enable your browser to execute JavaScript from your Jira applications to access their full functionality. You can consult the supported versions <a href="#">here</a> .
----------------	---

### Server-side requirements for evaluation purposes

<b>Java</b>	If you intend to use the <a href="#">Windows Installer</a> or <a href="#">Linux Installer</a> to install JIRA, there is no need to install and configure a separate JDK/JRE since these executable files will install and configure their own JRE to run JIRA, otherwise you will have to install a supported version of the ORACLE Java runtime. Consult the supported versions <a href="#">here</a> .
<b>Memory</b>	2 GB of Java heap size is enough for most evaluation purposes.
<b>Database</b>	Jira applications come pre-configured with the H2 database, which is suitable for evaluation purposes only, it shouldn't be used in production environments.
<b>Security</b>	Symantec must be uninstalled from the server that you want to install Jira applications on, as it is known to dramatically reduce application performance. For more information, see this knowledge base article: <a href="#">Crashes and Performance Issues Troubleshooting</a> .

## Server-side requirements for production

<b>Java</b>	If you intend to use the <a href="#">Windows Installer</a> or <a href="#">Linux Installer</a> to install Jira, there is no need to install and configure a separate JDK/JRE since these executable files will install and configure their own JRE to run Jira, otherwise you will have to install a supported version of the ORACLE Java runtime. Consult the supported versions <a href="#">here</a> .
<b>Hardware</b>	<ul style="list-style-type: none"> <li>• For a small number of projects (less or equal to 100) with 1,000 to 5,000 issues in total and about 100-200 users, a recent server (multicore CPU) with 8GB of available RAM and a reasonably fast hard drive (7200 rpm or faster) should cater for your needs. However, if you're using a 32-bit operating system, you shouldn't allocate more than 1GB of RAM to Jira. If you're manually installing/upgrading Jira on a 32-bit system by using the archive, you need to decrease the maximum heap size available to Jira. See the <a href="#">upgrade notes</a> for more information.</li> <li>• For 100 projects or more you should monitor Jira memory usage and allocate more memory if required. This is because each created project can create new workflows, new custom fields, new permissions schemes, new screens, etc.</li> <li>• If your system will experience a large number of concurrent requests, running Jira applications on a multicore CPU machine will increase the concurrency of processing the requests, and therefore, speed up the response time for your users.</li> <li>• For reference, we have a server that has a 2 Intel(R) Xeon(R) CPU E5520 @ 2.27 GHz (16 logical cores) with 32GB of RAM. This server runs Apache, various monitoring systems, and two Jira application instances: <ul style="list-style-type: none"> <li>• Our <a href="#">public site</a> has approximately: 145,000 issues, 255,000 comments, 120 custom fields, and 115 projects.</li> <li>• Our <a href="#">support site</a> has approximately: 285,000 issues, 2,500,000 comments, 75 custom fields, and 22 projects.</li> </ul> </li> </ul> <p>For more information, you can also refer to <a href="#">Scaling Jira</a>.</p>
<b>Database</b>	Using the embedded H2 database is not supported in production. You must install and connect your Jira instance to an <a href="#">enterprise database supported by Atlassian</a> .
<b>Security</b>	Symantec must be uninstalled from the server that you want to install Jira applications on, as it is known to dramatically reduce application performance. For more information, see this knowledge base article: <a href="#">Crashes and Performance Issues Troubleshooting</a> .

 If you are considering running Jira applications on VMware, please read [Virtualizing Jira \(Jira on VMware\)](#).

## Next Steps

[Installing Jira applications](#)

# Installing Java

Here you will find instructions for installing the Java Development Kit (JDK). This is a manual step that's only required if you're installing a Jira application from a zip or archive file. If you're using the [Windows installer](#) or [Linux installer](#), you don't need to install Java manually.

Check the [Supported platforms](#) page to find which Java versions are supported for Jira.

## Installing Java

You'll need to install the JDK on the same server that will have your Jira application.

### On Linux and Mac OS X

Before you start, check whether a JDK is already installed.

1. Open a shell console and type `echo $JAVA_HOME` and hit **Enter**:
  - If it returns something like `/opt/JDK7` or `/usr/lib/jvm/java-7` then your JDK is installed and configured
  - If nothing displays, you'll need to install the JDK or set the `$JAVA_HOME` environment variable
2. Check the [Supported platforms](#) page to find out which JDK versions are supported for your version of Jira.
3. Download the appropriate [Oracle JDK](#) or [AdoptOpenJDK](#) version.
4. Run the Java installer. Detailed installation instructions are provided on <http://www.oracle.com/technetwork/java/javase/index-137561.html>.
5. Open a shell console and type `echo $JAVA_HOME` and hit **Enter** to check that it has installed correctly.

### On Windows

Before you start, check whether a JDK is already installed.

1. Go to **Control Panel > Programs and Features** to see what JDK version is installed.
2. Check the [Supported platforms](#) page to find out which JDK versions are supported for your version of Jira applications.
3. Download the right [Oracle JDK](#) or [AdoptOpenJDK](#) version.
4. Run the Java installer. Make a note of the installation directory, as you'll need this later.
5. Open a command prompt and type `echo %JAVA_HOME%` and hit **Enter**:
  - If you see a path to your Java installation directory, the `JAVA_HOME` environment variable has been set correctly.
  - If nothing is displayed, or only `%JAVA_HOME%` is returned, you'll need to set the `JAVA_HOME` environment variable manually.

## Set the JAVA\_Home

If you installed the JDK, you'll be setting the `JAVA_HOME` environment variable. If you installed the Java Runtime Environment (JRE), follow the same steps, but set the `JRE_HOME` environment variable instead.

### On Linux

The `JAVA_HOME` environment variable is sometimes set in the `/etc/environment` file. You'll need to modify its value to `JAVA_HOME="path/to/JAVA_HOME"`.

1. If `JAVA_HOME` is not defined in this file, set it using the following command at a shell prompt, when logged in with 'root' level permissions:

```
export JAVA_HOME="path/to/JAVA_HOME" >> /etc/environment
```

2. Log out for these changes to apply.

### On Mac OS X

The `JAVA_HOME` environment variable is set in the `~/.bash_profile` file. You'll need to modify its value to `JAVA_HOME="path/to/JAVA_HOME"`.

1. If `JAVA_HOME` is not defined in this file, set it using the following command at a shell prompt, when logged in with 'root' level permissions:

```
export JAVA_HOME="path/to/JAVA_HOME" >> ~/.bash_profile
```

2. You'll need to open a new terminal for these changes to apply.

## On Windows

1. Locate your Java installation directory, it will be something like `C:\Program Files\Java\jdk1.8.0_65`
2. Do one of the following:
  - a. **Windows 7** Right click **My Computer** and select **Properties > Advanced**
  - b. **Windows 8** Go to **Control Panel > System > Advanced System Settings**
  - c. **Windows 10** Search for **Environment Variables** then select **Edit the system environment variables**
3. Click the **Environment Variables** button.
4. Under **System Variables**, click **New**.
5. In the **Variable Name** field, enter:
  - `JAVA_HOME` if you installed the JDK
  - `JRE_HOME` if you installed the JRE
6. In the **Variable Value** field, enter your JDK or JRE installation path.

 For Windows users on 64-bit systems:

```
Progra~1 = 'Program Files'
Progra~2 = 'Program Files(x86)'
```

7. Click **OK** and **Apply Changes** as prompted.
8. You'll need to close and re-open any command windows that were open before you made these changes. If the changes don't take effect after reopening the command window, restart Windows.

If you start Jira and you get an error like **Windows cannot find '-Xms128m'** you've probably not set `JAVA_HOME` correctly.

# Supported platforms

Before installing Jira, make sure you have the right software and infrastructure to run it. If a platform and version is not listed on this page, it means we don't test it, fix bugs or provide assistance for it. All platforms are shared between Jira Server and Jira Data Center, unless they're clearly marked as Data Center only.

 **This page is for Jira 8.20** . If you're looking for a different version, select it at the top-right.

## Definitions:

 Supported - you can use **Jira 8.20** with this platform.

 Limited - you can evaluate **Jira 8.20** on this platform, but you can't run a production site on it.

 Deprecated - you can use **Jira 8.20** with this platform, but we're planning to end support in an upcoming release.

## Further information:

- Please read [Jira applications installation requirements](#), since not all the platforms listed below may be required for your specific Jira setup.

## Java

### Oracle JRE / JDK:

 Java 8

 Java 11

### OpenJDK:

 Java 8

 Java 11 (AdoptOpenJDK 11.0.11+9)

### Good to know:

#### Oracle JRE / JDK

- We recommend using the Critical Patch Update (CPU) releases.

#### OpenJDK / AdoptOpenJDK

- Our Support and Engineering teams use **AdoptOpenJDK** to replicate any issues raised with OpenJDK. If you're using a different distribution of OpenJDK (e.g. Zulu), we'll still provide support for our products. However, if the bug is caused by a problem in Java distribution, we'll ask you to reach out to the Java distributor for help.
- You don't need to install Java if using the [Windows Installer](#) or [Linux Installer](#) as **AdoptOpenJDK JRE** is bundled with Jira.

## Operating systems

### Operating systems:

 Microsoft Windows

 Linux

 Mac OS X

 Amazon Web Services (AWS)

 Microsoft Azure

### Good to know:

- Jira is a pure Java-based application and should run on any supported operating system, provided that the JDK / JRE requirements are satisfied.

#### Microsoft Windows:

- Read [Anti-Virus in Jira](#) applications.

#### Linux:

- We perform tests on Ubuntu.

**AWS / Azure:**

- If you're creating your own deployment in AWS or Azure, you can use any configuration (e.g. OS or database) that's supported both by Jira and AWS/Azure.
- If you're using our reference templates, you will deploy a pre-configured environment, with some level of customizations available. For more info on our templates, see [Jira Data Center on AWS](#) or [Jira Data Center on Azure](#).

**Browsers**

**Desktop browsers:**

- ✔ Chrome (latest stable version)
- ✔ Microsoft Edge (Chromium, latest stable version)
- ✔ Mozilla Firefox (latest stable version)
- ✔ Safari on Mac OS X only (latest stable version)

**Mobile browsers:**

- ✔ Chrome (latest stable version)
- ✔ Safari on iOS only (latest stable version)
- ✔ Android 4.0 (Ice Cream Sandwich)

**Good to know:**

- We support a minimum screen resolution of 1024 x 768 (when browsers are maximized).

**Mobile:**

- You can view Jira on a mobile device using the [Jira mobile app](#) or the mobile view (browser).

**Databases**

PostgreSQL:	
✔ PostgreSQL 12	Jira is tested and bundled with the 42.2.23 JDBC driver. You can also use the latest JDBC driver for your <a href="#">PostgreSQL version</a> , though we can't guarantee it will work with your version of Jira. To use a different JDBC driver:
✔ PostgreSQL 11	<ol style="list-style-type: none"> <li>1. Stop your Jira instance.</li> <li>2. Remove the bundled driver from &lt;JIRA_INST&gt;/lib/.</li> <li>3. Download the new driver and place it in &lt;JIRA_INST&gt;/lib/.</li> <li>4. Restart your Jira instance.</li> </ol>
✔ PostgreSQL 10	<div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p><b>Jira 8.x</b></p> <p>Jira does not support reducing the <i>blocksize</i> parameter below its default value.</p> </div>

**MySQL:**

<ul style="list-style-type: none"> <li>✔ MySQL 8.0</li> <li>✔ MySQL 5.7</li> </ul>	<ul style="list-style-type: none"> <li>• Jirawill not work on:             <ul style="list-style-type: none"> <li>• MariaDB norPerconaDB</li> </ul> </li> <li>• We recommendrunning MySQL in strict mode.</li> <li>• Supported driver:             <ul style="list-style-type: none"> <li>• MySQL 8.0: <a href="#">MySQL Connector/J 8.0 driver</a></li> <li>• MySQL 5.7:                 <ul style="list-style-type: none"> <li>• <a href="#">MySQL Connector/J 8.0 driver</a></li> <li>• <a href="#">MySQL Connector/J 5.1 driver</a></li> </ul> </li> </ul> </li> </ul> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p><b>ℹ Jira 8.x</b></p> <p>Jira does not support reducing the <i>innodb_page_size</i> parameter below its default value.</p> </div>
--	--

Oracle	
<ul style="list-style-type: none"> <li>✔ Oracle 19c</li> <li>✔ Oracle 18c</li> <li>✔ Oracle 12c R2</li> </ul>	<ul style="list-style-type: none"> <li>• Jirawill not work on:             <ul style="list-style-type: none"> <li>• Oracle Advanced Compression Option (ACO)</li> </ul> </li> <li>• <b>JDBC Driver:</b> For all Oracle versions, use the JDBC 19.3 (<i>ojdbc8</i>) driver <a href="#">listed here</a>. This is the driver were using to test Jira with Oracle.</li> </ul> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p><b>ℹ Jira 8.x</b></p> <p>Jira does not support reducing the <i>DB_BLOCK_SIZE</i> parameter below its default value.</p> </div>

Microsoft SQL Server	
<ul style="list-style-type: none"> <li>✔ SQL Server 2019</li> <li>✔ SQL Server 2017</li> <li>✔ SQL Server 2016</li> </ul>	<p>Jirais tested and bundled with version 9.2.1.jre8 of the Microsoft JDBC driver. You can also use <a href="#">the latest JDBC driver</a> for yourversion of Microsoft SQL Server,though we can't guarantee it will work with your version ofJira. To use a different JDBC driver:</p> <ol style="list-style-type: none"> <li>1. Stop yourJirainstance.</li> <li>2. Remove the bundled driver from &lt;JIRA_INST&gt;/lib/.</li> <li>3. Download the new driver and place it in &lt;JIRA_INST&gt;/lib/.</li> <li>4. Restart yourJirainstance.</li> </ol> <div style="border: 1px solid #ffc107; padding: 5px; margin-top: 10px;"> <p><b>⚠</b> Jirawill not work on Express Editions of Microsoft SQL Server.</p> </div> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p><b>ℹ Jira 8.x</b></p> <p>Jira does not support reducing the <i>blocksize</i> parameter below its default value.</p> </div>

<b>Microsoft Azure</b>
------------------------

<ul style="list-style-type: none"> <li>✓ Azure SQL /Azure SQL Database Managed Instance</li> </ul>	<ul style="list-style-type: none"> <li>• Azure SQL is supported both for Jira Server and Jira Data Center. For more info, see <a href="#">Connecting Jira to Azure SQL</a>.</li> </ul>
<b>Amazon Aurora (Data Center only)</b>	
<ul style="list-style-type: none"> <li>✓ Postgr eSQL 12</li> <li>✓ Postgr eSQL 11</li> <li>✓ Postgre SQL 10</li> </ul>	<ul style="list-style-type: none"> <li>• The only supported Amazon Aurora config is a PostgreSQL-compatible clustered database with one writer replicating to zero or more readers.<a href="#">Learn more</a></li> </ul>
<b>Embedded database</b>	
<ul style="list-style-type: none"> <li>ⓘ H2 1.4.200 (evaluation only)</li> </ul>	<ul style="list-style-type: none"> <li>• Jiraships with a built-in database (H2).</li> </ul>

## Environment and Infrastructure

### Containerization

- You can use [official templates](#) to deploy Jira in a Docker container, or customize a Docker deployment on your own.
- We support the Jira templates and can help with Jira-related problems, but we don't provide support for Docker itself.

### Hardware

- NFS mounts are supported only for Jira Data Center's shared home directory. They won't work for Server, or DC's local home directory due to Lucene requirements. Read the [IndexWriter docs](#) for more information.
- We only support Jira running on x86 hardware and 64-bit derivatives of x86 hardware.
- If you are installing Jira from an archive, create a dedicated user account on the operating system to run Jira, since Jira runs as the user it is invoked under, it can potentially be abused.

### Virtualization

- VMware supports all of the operating systems listed under 'Operating systems'.
- We don't provide support for VMWare itself.
- Read [Virtualizing Jira](#) for information on how to configure VMWare.

### Application server

- We support Apache Tomcat 8.5.65.
- We don't support deploying multiple Atlassian applications in a single Tomcat container.

### Internet protocols (IP)

- We support IPv4.
- We support IPv6 with some limitations. See [IPv6 in Jira](#).

### External user directories

You can manage users in the following LDAP directories:

- Apache Directory Server 1.0.x
- Apache Directory Server 1.5.x

- Apple Open Directory (Read-Only)
- FedoraDS (Read-Only Posix Schema)
- Generic Directory Server
- Generic Posix/RFC2307 Directory (Read-Only)
- Microsoft Active Directory
- Novell eDirectory Server
- OpenDS
- Open LDAP
- Open LDAP (Read-Only Posix Schema)
- Sun Directory Server Enterprise Edition

### **Mail servers**

- SMTP servers must be able to support the multipart content type.

# End of support announcements

This page contains announcements of the end of support for various platforms and browsers used with Jira. These are summarized for upcoming Jira releases in the table below. Please see the following sections for the full announcements.

[End of support matrix for Jira](#)

The table below summarizes the end of support announcements for **upcoming** Jira releases.

Platform/Functionality	Jira end of support
PostgreSQL 9.6	From Jira 8.19 ( <a href="#">announcement</a> )

## Why is Atlassian ending support for these platforms?

Atlassian is committed to delivering improvements and bug fixes as fast as possible. We are also committed to providing world class support for all the platforms our customers run our software on. However, as new versions of databases, web browsers, etc, are released, the cost of supporting multiple platforms grows exponentially, making it harder to provide the level of support our customers have come to expect from us. Therefore, we no longer support platform versions marked as end-of-life by the vendor, or very old versions that are no longer widely used.

## End of support for PostgreSQL 9.6

*Announced May 2021*

**Platform:** PostgreSQL 9.6

**End of support:** Jira 8.19

In Jira 8.19, we'll be permanently removing support for PostgreSQL 9.6. End of support means that Atlassian will not fix bugs related to this platform in Jira 8.19 and later.

We're making this decision to reduce our database testing and support, and help us speed up our ability to deliver market-driver features. If you have questions or concerns regarding this announcement, please email [ol-announcement@atlassian.com](mailto:ol-announcement@atlassian.com).

Platform/Functionality	Jira end of support
MySQL 5.6	From Jira 8.12 ( <a href="#">announcement</a> )
SQL Server 2014	From Jira 8.12 ( <a href="#">announcement</a> )
Hipchat	From Jira 8.11 ( <a href="#">announcement</a> )
Oracle 12c R1	From Jira 8.8 ( <a href="#">announcement</a> )
Solaris	From Jira 8.8 ( <a href="#">announcement</a> )
PostgreSQL 9.4, 9.5	From Jira 8.8 ( <a href="#">announcement</a> )
SQL Server 2012	From Jira 8.8 ( <a href="#">announcement</a> )
Internet Explorer 11	From Jira 8.6 ( <a href="#">announcement</a> )
Built-in importers	From Jira 8.4 ( <a href="#">announcement</a> )
32-bit installers	From Jira 8.2 ( <a href="#">announcement</a> )
Jira CDN dark feature	From Jira 8.2 ( <a href="#">announcement</a> )

PostgreSQL 9.3	From Jira 8.0 ( <a href="#">announcement</a> )
com.atlassian.fugue	From Jira Service Desk 4.0 ( <a href="#">announcement</a> )
MySQL 5.5	From Jira 8.0 ( <a href="#">announcement</a> )
Postgres 9.2	From Jira 7.4 ( <a href="#">announcement</a> )
Internet Explorer 10	From Jira 7.2 ( <a href="#">announcement</a> )
Microsoft SQL Server 2008	From Jira 7.2 ( <a href="#">announcement</a> )
Postgres 9.0 and 9.1	From Jira 7.2 ( <a href="#">announcement</a> )
MySQL 5.1	From Jira 7.2 ( <a href="#">announcement</a> )
HSQLDB	From Jira 7.0 ( <a href="#">announcement</a> )
Oracle JDK 1.7	From Jira 7.0 ( <a href="#">announcement</a> )
Oracle 11G	From Jira 7.0 ( <a href="#">announcement</a> )
Internet Explorer 9	From Jira 7.0 ( <a href="#">announcement</a> )
SOAP API (replaced with REST)	From Jira 7.0 ( <a href="#">announcement</a> )
Jelly script	From Jira 6.4 ( <a href="#">announcement</a> )
WAR download distribution	From Jira 7.0 ( <a href="#">announcement</a> )
Microsoft SQL Server 2005	From Jira 7.0 ( <a href="#">announcement</a> )

## End of support for MySQL 5.6

*Announced December 2019*

**Platform:**MySQL 5.6

**End of support:**Jira 8.12

In Jira 8.12, we'll be permanently removing the support for MySQL 5.6. End of support means that Atlassian will not fix bugs related to these platforms in Jira 8.12 and later.

We're making this decision to reduce our database testing and support, and help us speed up our ability to deliver market-driver features. If you have questions or concerns regarding this announcement, please email [eol-announcement@atlassian.com](mailto:eol-announcement@atlassian.com).

## End of support for SQL Server 2014

*Announced December 2019*

**Platform:**SQL Server 2014

**End of support:**Jira 8.12

In Jira 8.12, we'll be permanently removing the support for Microsoft SQL Server 2014. End of support means that Atlassian will not fix bugs related to these platforms in Jira 8.12 and later.

We're making this decision to reduce our database testing and support, and help us speed up our ability to deliver market-driver features. If you have questions or concerns regarding this announcement, please email [ol-announcement@atlassian.com](mailto:ol-announcement@atlassian.com).

## End of support for Hipchat

*Announced March 2020*

**Solution:** Hipchat Server

**End of support:** Jira 8.11

We are planning to stop supporting and unbundle the Hipchat plugin from Jira version 8.11 and later. The reason for this is that Hipchat Cloud reached the end of life in Feb 2019, HipChat Data Center in Sep 2019, and Hipchat Server is due in June 2020.

## End of support for Oracle 12c R1

*Announced December 2019*

**Platform:** Oracle 12c R1

**End of support:** Jira 8.8

In Jira 8.8, we'll be permanently removing the support for Oracle 12c R1. End of support means that Atlassian will not fix bugs related to these platforms in Jira 8.8 and later.

We're making this decision to reduce our database testing and support, and help us speed up our ability to deliver market-driver features. If you have questions or concerns regarding this announcement, please email [ol-announcement@atlassian.com](mailto:ol-announcement@atlassian.com).

## End of support for Solaris

*Announced December 2019*

**Platform:** Solaris

**End of support:** Jira 8.8

In Jira 8.8, we'll be permanently removing the support for the Solaris operating system. End of support means that Atlassian will not fix bugs related to these platforms in Jira 8.8 and later.

If you have questions or concerns regarding this announcement, please email [ol-announcement@atlassian.com](mailto:ol-announcement@atlassian.com).

## End of support for PostgreSQL 9.4, 9.5

*Announced December 2019*

**Platform:** PostgreSQL 9.4, PostgreSQL 9.5

**End of support:** Jira 8.8

In Jira 8.8, we'll be permanently removing the support for PostgreSQL 9.4 and 9.5. End of support means that Atlassian will not fix bugs related to these platforms in Jira 8.8 and later.

We're making this decision to reduce our database testing and support, and help us speed up our ability to deliver market-driver features. If you have questions or concerns regarding this announcement, please email [eol-announcement@atlassian.com](mailto:eol-announcement@atlassian.com).

## End of support for SQL Server 2012

*Announced December 2019*

**Platform:** SQL Server 2012

**End of support:** Jira 8.8

In Jira 8.8, we'll be permanently removing the support for Microsoft SQL Server 2012. End of support means that Atlassian will not fix bugs related to these platforms in Jira 8.8 and later.

We're making this decision to reduce our database testing and support, and help us speed up our ability to deliver market-driver features. If you have questions or concerns regarding this announcement, please email [eol-announcement@atlassian.com](mailto:eol-announcement@atlassian.com).

## Deprecated browsers for Jira

In 2015 Microsoft released Edge as the browser to supersede Internet Explorer, and in recent times [Microsoft has discouraged the use of Internet Explorer as a default browser](#). To allow us to continue to take advantage of modern web standards to deliver improved functionality and the best possible user experience across all of our products, we have decided to end support for Internet Explorer 11.

End of support means we will not fix bugs specific to Internet Explorer 11, and will begin to introduce features that aren't compatible with this browser.

### When is this happening?

- Jira 8.5 will be the last version to support Internet Explorer 11.
- Subsequent versions will not support Internet Explorer 11.

### What this means for you

We recommend switching to one of our [supported browsers](#), such as Microsoft Edge, Google Chrome, or Mozilla Firefox.

If you have questions or concerns regarding this announcement, please email [eol-announcement@atlassian.com](mailto:eol-announcement@atlassian.com).

## End of support for several built-in importers

*Announced April 2019*

Jira allows you to import data from other issue trackers, like Asana, Bugzilla, or Pivotal Tracker. These imports are handled by the Jira Importers plugin, which is bundled with Jira and can be accessed by going to  > **System** > **External System Import**.

In Jira 8.4, we will remove import paths that are dedicated to specific issue trackers, leaving only generic paths that let you import data in the CSV and JSON format. You won't be able to go through a customized import path for e.g. Asana or Pivotal Tracker, but if these applications (and any other) allow to export data into CSV/JSON, you will still be able to import it to Jira.

Here's a list of applications and import paths that will be removed from **External System Import** in Jira 8.4:

- [Importing data from Excel](#)
- [Importing data from Bitbucket](#)

- [Importing data from Github](#)
- [Importing data from Asana](#)
- [Importing data from TFS or Visual Studio](#)
- [Importing data from Rally](#)
- [Importing data from VersionOne](#)
- [Importing data from YouTrack](#)
- [Importing data from Axosoft](#)
- [Importing data from Pivotal Tracker](#)
- [Importing data from Bugzilla](#)
- [Importing data from FogBugz On Demand](#)
- [Importing data from FogBugz for your Server](#)
- [Importing data from Mantis](#)
- [Importing data from Trac](#)
- [Importing data from Redmine](#)
- [Importing data from BaseCamp](#)

## End of support for 32-bit installers

*Announced April 2019*

The 32-bit installers have been removed in Jira 8.2. If you need to stick to 32-bit systems, you can still install Jira by using the `zip/tar.gz` archives.

## Removing Jira CDN dark feature

*Announced April 2019*

The unsupported Jira CDN dark feature `jira.fixed.cdn.enable` and system property `jira.fixed.cdn.prefix` will be removed in an upcoming Jira version. We recommend you turn off this dark feature, and remove it and the system property from your `setenv.sh` or `setenv.bat` file.

At the same time of removing this dark feature, we'll replace it with official CDN support for Jira Data Center, so stay tuned for details.

## End of support for PostgreSQL 9.3

*Announced August 2018*

Jira 7.12 has deprecated the use of PostgreSQL 9.3. In Jira 8.0, we'll be permanently removing the support for PostgreSQL 9.3. End of support means that Atlassian will not fix bugs related to PostgreSQL 9.3 in JIRA version 8.0 and later.

We are making this decision in order to reduce our database testing and support, and help us speed up our ability to deliver market-driven features. If you have questions or concerns regarding this announcement, please email `ol-announcement@atlassian.com`.

### PostgreSQL 9.3 deprecation notes:

- Jira 7.12 will be the version of Jira to officially deprecate PostgreSQL 9.3.
- Jira 7.x versions will continue to work with PostgreSQL 9.3.
- Jira 8.0 will not be tested against PostgreSQL 9.3.
- PostgreSQL 9.4 and 9.5 will continue to be supported in Jira 8.0 (see [Supported platforms](#)).

## End of support for `com.atlassian.fugue` in Jira Service Desk 4.0

*Announced August 2018*

Jira Service Desk 3.15 has deprecated the use of `com.atlassian.fugue`. In Jira Service Desk 4.0, we'll be permanently removing `com.atlassian.fugue` and updating our APIs to use **Core Java Data types and Exceptions** instead. Read the [full deprecation notice](#) for more information and post any questions in the [Atlassian Developer Community](#).

## End of support for MySQL 5.5

*Announced July 2018*

Atlassian will end support for MySQL 5.5 in **Jira 8.0**. End of support means that Atlassian will not fix bugs related to MySQL 5.5 past the support end date.

We are making this decision in order to reduce our database testing and support, and help us speed up our ability to deliver market-driven features. If you have questions or concerns regarding this announcement, please email [eol-announcement@atlassian.com](mailto:eol-announcement@atlassian.com).

### MySQL 5.5 End of Support Notes:

- Jira 7.x versions will continue to work with MySQL 5.5.
- Jira 8.0 will not be tested against MySQL 5.5.
- MySQL 5.6 and 5.7 will continue to be supported in Jira 8.0.

## End of support for Postgres 9.2

*Announced May 2017*

Atlassian will end support for Postgres 9.2 in **Jira 7.4**. End of support means that Atlassian will not fix bugs related to Postgres 9.2 past the support end date.

We are making this decision in order to reduce our database testing and support, and help us speed up our ability to deliver market-driven features. If you have questions or concerns regarding this announcement, please email [eol-announcement@atlassian.com](mailto:eol-announcement@atlassian.com).

### Postgres 9.2 End of Support Notes:

- Jira 7.3 will be the last major version of Jira to officially support Postgres 9.2.
- Jira 7.3.x and earlier versions should continue to work with Postgres 9.2.
- Jira 7.4 will not be tested against Postgres 9.2.
- Postgres 9.3, 9.4 and 9.5 will continue to be supported in Jira 7.4.x (see [Supported platforms](#)).

## End of support for Internet Explorer 10

*Announced May 2016*

Atlassian will end support for Internet Explorer 10 (IE10) in **Jira 7.2**. End of support means that Atlassian will not fix bugs related to IE10 past the support end date.

We are making this decision in order to reduce our browser testing and support, and help us speed up our ability to deliver market-driven features. If you have questions or concerns regarding this announcement, please email [eol-announcement@atlassian.com](mailto:eol-announcement@atlassian.com).

### Internet Explorer 10 (IE10) End of Support Notes:

- Jira 7.1 will be the last major version of Jira to officially support IE10.
- Jira 7.1.x and earlier versions should continue to work with IE10.
- Jira 7.2 will not be tested against IE10.
- Internet Explorer 11 will continue to be supported in Jira 7.2.x (see [Supported platforms](#)).

## End of support for Microsoft SQL Server 2008

*Announced October 2015*

Atlassian will end support for Microsoft SQL Server 2008 in **Jira 7.2**. End of support means that Atlassian will not fix bugs related to Microsoft SQL Server 2008 past the support end date.

We are making this decision in order to reduce our database testing and support, and help us speed up our ability to deliver market-driven features. If you have questions or concerns regarding this announcement, please email [eol-announcement@atlassian.com](mailto:eol-announcement@atlassian.com).

### Microsoft SQL Server 2008 End of Support Notes:

- Jira 7.1 will be the last major version of Jira to officially support Microsoft SQL Server 2008.
- Jira 7.1.x and earlier versions should continue to work with Microsoft SQL Server 2008.
- Jira 7.2 will not be tested against Microsoft SQL Server 2008.
- Microsoft SQL Server 2012 will continue to be supported in Jira 7.2.x (see [Supported platforms](#)).

## End of support for Postgres 9.0 and 9.1

*Announced October 2015*

Atlassian will end support for Postgres 9.0 and Postgres 9.1 in **Jira 7.2**. End of support means that Atlassian will not fix bugs related to Postgres 9.0 or 9.1 past the support end date.

We are making this decision in order to reduce our database testing and support, and help us speed up our ability to deliver market-driven features. If you have questions or concerns regarding this announcement, please email [eol-announcement@atlassian.com](mailto:eol-announcement@atlassian.com).

### Postgres 9.0 and Postgres 9.1 End of Support Notes:

- Jira 7.1 will be the last major version of Jira to officially support Postgres 9.0 and 9.1.
- Jira 7.1.x and earlier versions should continue to work with Postgres 9.0 and 9.1.
- Jira 7.2 will not be tested against Postgres 9.0 or 9.1.
- Postgres 9.2 and Postgres 9.3 will continue to be supported in Jira 7.2.x (see [Supported platforms](#)).

## End of support for MySQL 5.1

*Announced October 2015*

Atlassian will end support for MySQL 5.1 in **Jira 7.2**. End of support means that Atlassian will not fix bugs related to MySQL 5.1 past the support end date.

We are making this decision in order to reduce our database testing and support, and help us speed up our ability to deliver market-driven features. If you have questions or concerns regarding this announcement, please email [eol-announcement@atlassian.com](mailto:eol-announcement@atlassian.com).

### MySQL 5.1, 5.2, 5.3 and 5.4 End of Support Notes:

- Jira 7.1 will be the last major version of Jira to officially support MySQL 5.1.
- Jira 7.1.x and earlier versions should continue to work with MySQL 5.1.
- Jira 7.2 will not be tested against MySQL 5.1.
- MySQL 5.5 and 5.6 will continue to be supported in Jira 7.2.x (see [Supported platforms](#)).

## End of support for HSQLDB

*Announced February 2015*

Atlassian will end support for HSQLDB (HyperSQL DataBase) in **Jira 7.0**. End of support means that Atlassian will not fix bugs in HSQLDB past the support end date.

Jira ships with a built-in database for evaluation purposes, and currently this is HSQLDB. As of **Jira 7.0**, Jira will ship with H2 (H2 Database Engine) as its built-in database.

#### **HSQLDB (HyperSQL DataBase or HSQLDB) End of Support Notes:**

- Jira 6.4 will be the last major version of Jira to officially support HSQLDB (HyperSQL DataBase) for evaluation use.
- Jira 6.4.x and earlier versions will continue to work with HSQLDB (HyperSQL DataBase) for evaluation use. However, we will not fix bugs affecting HSQLDB (HyperSQL DataBase) past the support end date.
- Jira 7.0 will not be tested with HSQLDB (HyperSQL DataBase).

## End of support for Oracle JDK 1.7

*Announced February 2015*

Atlassian will end support for Java 7 (JRE and JDK 1.7) in **Jira 7.0**. End of support means that Atlassian will not fix bugs in Java 7 (JRE and JDK 1.7) past the support end date.

We are ending support for Java 7 (JRE and JDK 1.7), as Oracle Corporation has announced the end of public updates for Java 7: [Java SE 7 End of Public Updates Notice](#).

#### **Java 7 (JRE and JDK 1.7) End of Support Notes:**

- Jira 6.4 will be the last major version of Jira to officially support Java 7 (JRE and JDK 1.7).
- Jira 6.4.x and earlier versions will continue to work with Java 7 (JRE and JDK 1.7). However, we will not fix bugs affecting Java 7 (JRE and JDK 1.7) past the support end date.
- Jira 7.0 will not be tested with Java 7 (JRE and JDK 1.7).
- Java 8 (JRE and JDK 1.8) is supported, but not bundled with Jira 6.4

## End of support for Oracle 11G

*Announced February 2015*

Atlassian will end support for Oracle 11G in **Jira 7.0**. End of support means Atlassian will not fix bugs related to Oracle 11G past the support end date, except for security-related issues.

We are making this decision as Oracle Corporation have ended support for Oracle 11G as of [January 2015](#). Testing on Oracle 12C will conclude shortly and we'll announce support soon.

#### **Oracle 11G End of Support Notes**

- Jira 6.4 will be the last major release that supports Oracle 11G
- Jira 6.4.x and earlier versions will continue to work on Oracle 11G
- Jira 7.0 will not be tested against Oracle 11G

## End of support for Internet Explorer 9

*Announced February 2015*

Atlassian will end support for Internet Explorer 9 in **Jira 7.0**. End of support means that Atlassian will not fix bugs related to Internet Explorer 9 past the support end date, except for security-related issues.

We are making this decision to enable us to provide the best user experience to our customers, accelerate our pace of innovation, and give us the ability to utilize modern browser technologies.

#### **Internet Explorer 9 (IE9) End of Support Notes**

- Jira 6.4 will be the last major release that supports Internet Explorer 9
- Jira 6.4.x and earlier versions should continue to work on Internet Explorer 9

- Jira 7.0 will not be tested against Internet Explorer 9
- Internet Explorer 10 and Internet Explorer 11 will continue to be supported in Jira 7.0.x.

## End of support for SOAP

*Announced November 2014*

Atlassian will end support for SOAP API in **Jira 7.0**. The SOAP API's have been replaced by [REST API's](#) as Atlassian's recommended and supported remote API.

### SOAP End of Support Notes

- Jira 6.4 will be the last major release that supports SOAP
- Jira 6.4.x and earlier versions should continue to work with SOAP
- Jira 7.0 will not include any SOAP API's
- If you need an alternative that Atlassian supports, the [REST API](#) is fully supported by Jira.

## End of support for Jelly Scripts

*Announced November 2014*

Atlassian will end support for Jelly scripts in **Jira 6.4**. If you are using Jelly scripts with Jira, we suggest you move to [Groovy Script Runner](#) or utilize the [Jira Command Line Interface](#), which will provide you with more flexible options.

### Jelly Script End of Support Notes

- Jira 6.3 will be the last major release to support Jelly scripts
- Jira 6.3.x and earlier versions should continue to work fine with Jelly scripts
- Jira 6.4 will not include Jelly.
- If you need an alternative to Jelly scripts, [Groovy Script Runner](#) or the [Jira Command Line Interface](#) are the suggested alternatives that work with Jira.

## End of support for WAR distribution

*Announced August 2014*

Atlassian will stop releasing the WAR distribution of Jira in **Jira 7.0**.

### Why are we ending support for this?

- We are trying to reduce the amount of combinations and confusion around this for customers downloading a Server (BTF) edition
- The WAR edition is a bit more complex to install and gets more difficult as the installation ages and gets bigger - we want to reduce that complexity
- We can't and don't test every permutation of environments + app servers that a customer might deploy into, nor can we control what else might be in that environment, which can lead to a poor user experience
- We only support Tomcat - Jira doesn't work on WLS or WebSphere anyways, other app servers - maybe.

Anything we release, we want to make sure users get a good experience in installation and usage and don't have to deal with app server quirks etc.

## End of support for Microsoft SQL Server 2005

*Announced June 2014*

Atlassian will end support for Microsoft SQL Server 2005 in **Jira 7.0**. End of support means that Atlassian will not fix bugs related to Microsoft SQL Server 2005 past the support end date.

We are making this decision in order to reduce our database testing and support, and help us speed up our ability to deliver market-driven features. If you have questions or concerns regarding this announcement, please email [ol-announcement@atlassian.com](mailto:ol-announcement@atlassian.com).

**Microsoft SQL Server 2005 End of Support Notes:**

- Jira 6.4 will be the last major version of Jira to officially support Microsoft SQL Server 2005.
- Jira 6.4.x and earlier versions should continue to work with Microsoft SQL Server 2005.
- Jira 7.0 will not be tested against Microsoft SQL Server 2005.
- Microsoft SQL Server 2008 and 2008 R2 will continue to be supported in Jira 7.0.x (see [Supported Platforms](#)).
- We will start supporting Microsoft SQL Server 2012 in Jira 6.4.

# Install a Jira trial

Want to quickly get up and running with a Jira application? This page will guide you through a few simple steps to install and set up a trial Jira Data Center site. If you'd rather set up a production site or you just want more control, see [Installing Jira Data Center](#).

 We no longer offer Server evaluation licenses, but you can still trial our Data Center products. [Learn more about these changes](#)

## Before you begin

Jira installers come with all the bits and pieces you need to run the application, but there's a few things you'll need to get up and running:

 For the list of supported platforms, see [Supported platforms](#).

- **Operating system:** A computer or laptop with a supported operating system you'll be installing Jira so you'll need admin rights.

You can install Jira on a Windows or Linux operating system.

Apple Mac isn't supported for production sites, but if you're comfortable setting up applications on your Mac from scratch, you can download the `tar.gz` file and follow the instructions for [Installing Jira applications on Linux from Archive File](#) as the process is similar.

- **Database:** You don't need a database if you're planning to evaluate Jira Data Center on a single node (it comes with a built-in H2 database). To evaluate clustering features, you'll need an external database, so your nodes can access it.
- **Web browser:** You'll need this to access Jira.
- **Email address:** You'll need this to generate your evaluation license and create an account.

Ready to get going? Let's start with grabbing the installer.

## 1. Download the installer

Download the installer for your operating system:

- **Jira Software:** <https://www.atlassian.com/software/jira/download>
- **Jira Service Management:** <https://www.atlassian.com/software/jira/service-desk/download>

## 2. Install your Jira application

The installer allows you to choose Express or Custom installations. The Custom installation allows you to pick some specific options for Jira, but for this guide we'll use the Express installation.

1. Run the installer, we recommend running with a Windows administrator account. If prompted, make sure you allow the installer to make changes to your computer. This will allow you to install Jira as a service.
  2. Choose **Express Install**, then click **Next**.
  3. Once installation is complete, it will ask you if you want to open Jira in your browser. Make sure this option is selected then click **Done**.
  4. Jira will open in your default browser, and you're ready to start the set up wizard.
1. Change to the directory where you downloaded Jira then execute this command to make it executable:

<b>Jira Core</b>
<pre>\$ chmod a+x atlassian-jira-core-X.X.X-x64.bin</pre>
<b>Jira Software</b>
<pre>\$ chmod a+x atlassian-jira-software-X.X.X-x64.bin</pre>
<b>Jira Service Desk</b>
<pre>\$ chmod a+x atlassian-servicedesk-X.X.X-x64.bin</pre>

Where `-X.X.X` is the version you downloaded.

2. Run the installer, we recommend using `sudo` to run the installer as this will create a dedicated account to run Jira and allow you to run Jira as a service.

<b>Jira Core</b>
<pre>\$ sudo ./atlassian-jira-core-X.X.X-x64.bin</pre>
<b>Jira Software</b>
<pre>\$ sudo ./atlassian-jira-software-X.X.X-x64.bin</pre>
<b>Jira Service Desk</b>
<pre>\$ sudo ./atlassian-servicedesk-X.X.X-x64.bin</pre>

3. When prompted, choose **Express Install** (option 1).
4. Once installation is complete head to <http://localhost:8080> in your browser to begin the setup process.

### 3. Set up your Jira application

The setup wizard is the last step in getting Jira up and running. You'll need your email address to generate your evaluation license.

1. **Single node:** If you're installing on a single node, select **Set it up for me**. This will allow Jira to set up everything it needs to run, including an H2 database.
2. **Cluster:** If you're installing in a cluster, select **I'll set it up myself**. This will allow you to provide connection details to your own database.
3. Create an account, or log in with your Atlassian ID account.
4. Follow the prompts to **generate a license** for the Jira application you want to try, and apply it to your new installation.
5. Enter and confirm the details you want to use for your administrator account, and click **Next**.

It will take a few minutes to get everything connected and operational. Once that's done, you're ready to go!

# Installing Jira applications on Windows

In this guide we'll run you through installing a Jira application in a production environment, with an external database, using the Windows installer.

This is the most straightforward way to get your production site up and running on a Windows server.



## Other ways to install Jira:

- [Evaluation](#)-get your free trial up and running in no time.
- [Zip](#) install Jira manually from a zip file.
- [Linux](#) install Jira on a Linux operating system

## On this page:

- [Before you begin](#)
- [Install a Jira application](#)
  - [1. Download Jira](#)
  - [2. Run the installer](#)
- [Set up your Jira application](#)
  - [3. Choose set up method](#)
  - [4. Connect to your database](#)
  - [5. Set application properties](#)
  - [6. Enter your license](#)
  - [7. Create your administrator account](#)
  - [8. Set up email notifications](#)
  - [9. Start using Jira](#)
- [Troubleshooting](#)

## Before you begin

Before you install Jira, there's a few questions you need to answer.

Are you using a supported operating system?	<p>Check the <a href="#">Supported platforms</a> page for the version of Jira you are installing. This will give you info on supported operating systems, databases and browsers.</p> <p><b>Good to know:</b></p> <ul style="list-style-type: none"><li>• We don't support installing Jira on OSX or mac OS.</li><li>• The Jira installer includes Java (JRE) and Tomcat, so you don't need to install these separately.</li></ul>
---	--

<p>Do you want to run Jira as a Windows Service?</p>	<p>Running Jira as a service in Windows means that your Jira application will automatically start up when Windows is started.</p> <p><b>If you choose to run Jira as a service:</b></p> <ul style="list-style-type: none"> <li>• You must run the installer as administrator to be able to install Jira as a service.</li> <li>• The Jira service will be run as the Windows 'SYSTEM' user account.</li> <li>• We strongly recommend creating a dedicated user account (e.g. with username 'jira') on Windows for running Jira.</li> </ul> <p>See <a href="#">Running Jira applications as a Window's service</a> for more information.</p> <p><b>If you choose not to run Jira as a service:</b></p> <ul style="list-style-type: none"> <li>• You will start and stop Jira by running the <code>start-jira.bat</code> file in your Jira installation directory.</li> <li>• Jira will be run as the Windows user account that was used to install Jira, or you can choose to run as a dedicated user.</li> <li>• Jira will need to be restarted manually if your server is restarted.</li> </ul>
<p>Is your database set up and ready to use?</p>	<p>To run Jira in production you'll need an external database. Check the <a href="#">Supported platforms</a> page for the version you're installing for the list of databases we currently support. If you don't already have a database, PostgreSQL is free, easy to set up and has been extensively tested with Jira.</p> <p><b>Good to know:</b></p> <ul style="list-style-type: none"> <li>▪ Set up your database before you begin. Step-by-step guides for all supported databases are available in <a href="#">Connecting Jira applications to a database</a>.</li> <li>▪ Use UTF-8 character encoding.</li> <li>▪ If you're using Oracle or MySQL you'll need to download the driver for your database.</li> <li>▪ The embedded H2 database can be used for evaluating Jira, but you'll need to migrate to another database before running in production. You may find it easier to use external database from the start.</li> </ul>
<p>Do you have a Jira license?</p>	<p>You'll need a valid Jira Software Server, Jira Core Server or Jira Service Desk Server license to use JIRA.</p> <p><b>Good to know:</b></p> <ul style="list-style-type: none"> <li>• If you have not yet purchased a Jira application license you'll be able to create an evaluation license during setup.</li> <li>• If you already have a license key you'll be prompted to log in to <a href="https://my.atlassian.com">my.atlassian.com</a> to retrieve it, or you can enter the key manually during setup.</li> <li>• If you're migrating from Jira Cloud, you'll need a new license for Jira.</li> </ul>

## Install a Jira application

### 1. Download Jira

Download the installer for your operating system:

- Jira Core at <https://www.atlassian.com/software/jira/core/download>
- Jira Software at <https://www.atlassian.com/software/jira/download>
- Jira Service Management at <https://www.atlassian.com/software/jira/service-desk/download>

## 2. Run the installer

1. Run the installer. We recommend using a Windows administrator account.
2. Follow the prompts to install Jira. You'll be asked for the following info:
  - a. **Destination directory** this is where Jira will be installed.
  - b. **Home directory** this is where Jira data like logs, search indexes and files will be stored.
  - c. **TCP ports** these are the HTTP connector port and control port Jira will run on. Stick with the default unless you're running another application on the same port.
  - d. **Install as service** this option is only available if you ran the installer as administrator.
3. Jira will start up in your browser once installation is complete.

Set up your Jira application

## 3. Choose set up method

Choose **I'll set it up myself**.

## 4. Connect to your database

1. If you've not already done so, it's time to create your database. See the 'Before you begin' section of this page for details.
2. Choose **My own database**.
3. Choose your database type then enter the details for your database.

JIRA connects to your database using a standard JDBC database connection. Connection pooling is handled within JIRA, you can change this using [JIRA configuration tool](#) later.

### If you're using Oracle or MySQL there's an extra step:

- Download and extract the appropriate database JDBC drivers. See [Supported platforms](#) to get the right version.
- Drop the JAR file into your `<jira-installation>/lib` folder before continuing with the setup wizard.

### In the setup wizard:

- **Driver Class Name** the Java class name for your database driver. If you're not sure, check the documentation for your database.
- **Database URL** the JDBC URL for your database. If you're not sure, check the documentation for your database.
- **Username and Password** A valid username and password that JIRA can use to access your database.

## 5. Set application properties

1. Give your Jira site a name.
2. Choose whether your site should be private or anyone can sign up. You can change this later.
3. Enter your base URL - this is the address people will use to access your Jira site.

## 6. Enter your license

Follow the prompts to log in to [my.atlassian.com](https://my.atlassian.com) to retrieve your license, or enter a license key.

## 7. Create your administrator account

Enter details for the administrator account. You can add more administrators after set up is complete.

## 8. Set up email notifications

Enter details of your mail server. This will allow Jira to send notifications when issues change.

## 9. Start using Jira

That's it! Your Jirasite is accessible from your base URL or a URL like this: `http://<computer_name_or_IP_address>:<port>`

Here's a few things that will help you get your team up and running:

- [Add and invite users](#) to get your team on board, or [configure user directories](#) for slightly bigger teams.
- [Create your first project](#) to have something to work on.
- [Configure SSL or HTTPS](#) to keep Jira and your team more secure.

### Troubleshooting

Some anti-virus or other Internet security tools may interfere with the Jira installation process and prevent the process from completing successfully. If you experience or anticipate experiencing such an issue with your anti-virus/Internet security tool, disable this tool first before proceeding with the Jira installation.

Head to [Installation Troubleshooting](#) for more help.

# Uninstalling Jira applications from Windows

This page describes the procedure for uninstalling Jira, which had been installed using the [Windows Installer](#).

**i** If you wish to re-install Jira in 'unattended mode', do not uninstall your previous installation of Jira just yet. See [Using the silent installation feature](#) for more information.

## To uninstall Jira from Windows:

1. Log in to Windows as the same user that was used to install Jira with the [Windows Installer](#).
2. Start the uninstaller by doing one of the following:
  - Click the Windows 'Start' menu > 'All Programs' > 'JIRAX.Y' > 'Uninstall JIRAX.Y' (where 'X.Y' refers to the installed version of JIRA that you are about to uninstall)  
**OR**
  - Open the Windows Control Panel, choose 'Add or Remove Programs' (on Windows XP) or 'Programs and Features' on (Windows 7/Vista), and then uninstall 'JiraX.Y' from the list of applications  
**OR**
  - Open the Windows command prompt, and do the following:
    - a. Change directory `cd` to your Jira installation directory
    - b. Run the `uninstall.exe` file
3. Follow the prompts to uninstall Jira from your computer.

## **i** Please note:

- The uninstaller will not delete the Jira home directory.
- All log files that were generated while Jira was running will not be deleted.
- All files within the Jira installation directory will be deleted (with the exception of the Tomcat `log` folder located in the Jira installation directory).
- The uninstaller can be made to operate in unattended mode by specifying the `-q` option at the Windows command prompt i.e. `uninstall.exe -q`

# Installing Jira applications on Windows from Zip File

In this guide we'll run you through installing a Jira application in a production environment, with an external database, manually using a zip file.

This method gives you the most control of the installation process.



## Other ways to install Jira:

- [Evaluation](#) - get your free trial up and running in no time.
- [Installer](#) - install Jira using the Windows installer.
- [Linux](#) - install Jira on a Linux operating system.

## Before you begin

Before you install Jira, there's a few questions you need to answer.

### On this page:

#### [Before you begin](#)

#### [Install a Jira application](#)

1. [Download Jira](#)
2. [Create the installation directory](#)
3. [Create the home directory](#)
4. [Check the ports](#)
5. [Start Jira](#)

#### [Set up your Jira application](#)

6. [Choose set up method](#)
7. [Connect to your database](#)
8. [Set application properties](#)
9. [Enter your license](#)
10. [Create your administrator account](#)
11. [Set up email notifications](#)
12. [Start using Jira](#)

#### [Troubleshooting](#)

Are you using a supported operating system and Java version?

Check the [Supported platforms](#) page for the version of Jira you are installing. This will give you info on supported operating systems, databases and browsers.

### Good to know:

- We don't support installing Jira on OSX.
- You can use either the JDK (Java Development Kit) or JRE (Java Runtime Environment).
- We only support the version of Apache Tomcat that is bundled with Jira.

<p>Are you using a 32-bit operating system?</p>	<p>If you're installing Jira on a 32-bit system, you need to decrease the maximum heap size available to Jira. The default for 64-bit systems is 2GB, which is too much for a 32-bit system, and may not fit into the available memory.</p> <p>Complete these steps after extracting files from the archive, but before starting Jira.</p> <p><b>Step 1: Rename the default setenv file.</b></p> <ol style="list-style-type: none"> <li>1. Go to <code>&lt;Jira-install-directory&gt;/bin</code>, and delete the <code>setenv.bat / .sh</code> file (or change its name).</li> <li>2. Rename <code>setenv32.bat / .sh</code> to <code>setenv.bat / .sh</code>. Jira will use this file on startup.</li> </ol> <p><b>Step 2: Add the properties to the <code>jira-config.properties</code> file.</b></p> <ol style="list-style-type: none"> <li>1. Go to Jira's home directory, and edit the <code>jira-config.properties</code> file. If the file isn't there, you can <a href="#">create it</a>.</li> <li>2. Add the following properties:</li> </ol> <pre style="border: 1px solid #ccc; padding: 5px;">jira.index.batch.maxrambuffermb=256 jira.index.interactive.maxrambuffermb=256</pre>
<p>Do you want to run Jira as a Windows Service?</p>	<p>Running Jira as a service in Windows means that your Jira application will automatically start up when Windows is started.</p> <p>You should use the <a href="#">Windows installer</a> if you want to run Jira as a Service.</p> <p><b>If you choose not to run Jira as a service:</b></p> <ul style="list-style-type: none"> <li>• You will start and stop Jira by running the <code>start-jira.bat</code> file in your Jira installation directory.</li> <li>• Jira will be run as the Windows user account that was used to install Jira, or you can choose to run as a dedicated user (this user must have full read and write access to the installation directory and home directory).</li> <li>• Jira will need to be restarted manually if your server is restarted.</li> </ul>
<p>What database do you plan to use?</p>	<p>To run Jira in production you'll need an external database. Check the <a href="#">Supported platforms</a> page for the version you're installing for the list of databases we currently support. If you don't already have a database, PostgreSQL is free, easy to set up and has been extensively tested with Jira.</p> <p><b>Good to know:</b></p> <ul style="list-style-type: none"> <li>■ Set up your database before you begin. Step-by-step guides for all supported databases are available in <a href="#">Connecting Jira applications to a database</a>.</li> <li>■ Use UTF-8 character encoding.</li> <li>■ If you're using Oracle or MySQL you'll need to download the driver for your database.</li> <li>■ The embedded H2 database can be used for evaluating Jira, but you'll need to migrate to another database before running in production. You may find it easier to use external database from the start.</li> </ul>

Do you have a Jira license?	<p>You'll need a valid JiraSoftware Server, JiraCore Server or JiraService Desk Server license to use JIRA.</p> <p><b>Good to know:</b></p> <ul style="list-style-type: none"> <li>• If you have not yet purchased a Jiraapplication license you'll be able to create an evaluation license during setup.</li> <li>• If you already have a license key you'll be prompted to log in to <a href="https://my.atlassian.com">my.atlassian.com</a> to retrieve it, or you can enter the key manually during setup.</li> <li>• If you're migrating from JiraCloud, you'll need a new licenseJira</li> </ul>
Is your JAVA_HOME variable set correctly?	<p>Before you install Jira, check that you're running a supported Java version and that the JAVA_HOME environment variable is set correctly.</p> <p><b>Jira applications can run with OpenJDK, Oracle JDK or JRE.</b></p> <p>To check the JAVA_HOME variable:</p> <p>Open a command prompt and type <code>echo %JAVA_HOME%</code> and hit Enter.</p> <ul style="list-style-type: none"> <li>• If you see a path to your Java installation directory, the JAVA_Home environment variable has been set correctly.</li> <li>• If nothing is displayed, or only <code>%JAVA_HOME%</code> is returned, you'll need to set the JAVA_HOME environment variable manually.</li> </ul>

## Install a Jiraapplication

### 1. DownloadJira

Download the zip file for your operating system:

- Jira Core at <https://www.atlassian.com/software/jira/core/download>
- Jira Software at <https://www.atlassian.com/software/jira/download>
- Jira Service Managementat <https://www.atlassian.com/software/jira/service-desk/download>

### 2. Create the installation directory

1. Create your installation directory (with full control permission) this is where Jirawill be installed. Avoid using spaces or special characters in the path. We'll refer to this directory as your `<installation-directory>`.
2. Extract the Jirazip file to your `<installation-directory>`. We recommend using [7zip](#) or [Winzip](#).

### 3. Create the home directory

1. Create your home directory (with full control permission) this is where Jiradata like logs, search indexes and files will be stored. This should be seperate to your installation directory. We'll refer to this directory as your `<home-directory>`.
2. Tell Jirawhere to find your `<home-directory>` when it starts up. There are two ways to do this:

You can set an environment variable named `JIRA_HOME` in your operating system with the absolute path to your `<home-directory>`.

Open Command Prompt and execute the following:

```
set JIRA_HOME=X:\path\to\jira-home
```

where `x` is the drive where you created your `<home-directory>`.

You can then specify the command above in a script used to start Jira.

Edit<installation-directory>\atlassian-jira\WEB-INF\classes\jira-application.properties file in any text editor.

After `jira.home` add the absolute path to your home directory. You will need to escape the backslashes, for example:

```
jira.home=X:\\path\\to\\jira-home
```

If you define an UNC path you will need to double escape the leading backslash, for example:

```
jira.home=\\\\\\machinename\\path\\to\\jira-home
```

#### 4. Check the ports

By default Jira listens on port 8080. If you have another application running on your server that uses the same ports, you'll need to tell Jira to use a different port.

To change the ports:

1. Edit <installation-directory>\conf\server.xml
2. Change the **Server** port (8005) and the **Connector** port (8080) to free ports on your server.

In the example below we've changed the **Server** port to 5005 and the **Connector** port to 5050.

```
<Server port="5005" shutdown="SHUTDOWN">
...
  <Service name="Catalina">
    <Connector port="5050"
      maxThreads="150"
      minSpareThreads="25"
      connectionTimeout="20000"
      enableLookups="false"
      maxHttpHeaderSize="8192"
      protocol="HTTP/1.1"
      useBodyEncodingForURI="true"
      redirectPort="8443"
      acceptCount="100"
      disableUploadTimeout="true"/>
```

#### 5. Start Jira

1. Run <installation-directory>/bin/start-jira.bat to start the install process.

A command prompt will open. Closing this window will stop Jira.

2. Go to <http://localhost:8080/> to launch Jira in your browser (change the port if you've updated the Connector port).
  - If the command prompt window closes immediately, your JAVA\_HOME variable may not be set correctly.

Set up your Jira application

#### 6. Choose set up method

Choose **I'll set it up myself**.

## 7. Connect to your database

1. If you've not already done so, it's time to create your database. See the 'Before you begin' section of this page for details.
2. Choose **My own database**.
3. Choose your database type then enter the details for your database.

JIRA connects to your database using a standard JDBC database connection. Connection pooling is handled within JIRA, you can change this using [JIRA configuration tool](#) later.

### If you're using Oracle or MySQL there's an extra step:

- Download and extract the appropriate database JDBC drivers. See [Supported platforms](#) to get the right version.
- Drop the JAR file into your `<jira-installation>/lib` folder before continuing with the setup wizard.

### In the setup wizard:

- **Driver Class Name** the Java class name for your database driver. If you're not sure, check the documentation for your database.
- **Database URL** the JDBC URL for your database. If you're not sure, check the documentation for your database.
- **Username and Password** A valid username and password that JIRA can use to access your database.

## 8. Set application properties

1. Give your Jirasite a name.
2. Choose whether your site should be private or anyone can sign up. You can change this later.
3. Enter your base URL - this is the address people will use to access your Jirasite.

## 9. Enter your license

Follow the prompts to log in to [my.atlassian.com](http://my.atlassian.com) to retrieve your license, or enter a license key.

## 10. Create your administrator account

Enter details for the administrator account. You can add more administrators after set up is complete.

## 11. Set up email notifications

Enter details of your mail server. This will allow Jira to send notifications when issues change.

## 12. Start using Jira

That's it! Your Jirasite is accessible from your base URL or a URL like this: `http://<computer_name_or_IP_address>:<port>`

Here's a few things that will help you get your team up and running:

- [Add and invite users](#) to get your team on board, or [configure user directories](#) for slightly bigger teams.
- [Create your first project](#) to have something to work on.
- [Configure SSL or HTTPS](#) to keep Jira and your team more secure.

## Troubleshooting

- If your web browser window shows an error the first time you try to access Jira, wait for 30 seconds or so and then refresh the page.

- If the command prompt window closes immediately, your JAVA\_HOME variable may not be set correctly.

Head to [Installation Troubleshooting](#) in our Knowledge Base for more help.

# Installing Jira applications on Linux

In this guide we'll run you through installing a Jira application in a production environment, with an external database, using the Linux installer.

This is the most straightforward way to get your production site up and running on a Linux server.



## On this page:

[Before you begin](#)

[Install a Jira application](#)

1. [Download Jira](#)

2. [Run the installer](#)

[Set up your Jira application](#)

3. [Choose set up method](#)

4. [Connect to your database](#)

5. [Set application properties](#)

5. [Enter your license](#)

6. [Create your administrator account](#)

7. [Set up email notifications](#)

8. [Start using Jira](#)

[Troubleshooting](#)

## Other ways to install Jira:

- [Evaluation](#)-get your free trial up and running in no time.
- [TAR.GZ](#) install Jira manually from an archive file.
- [Windows](#) install Jira on a Windows server.

## Before you begin

Before you install Jira, there are a few questions you need to answer.

<p>Are you using a supported operating system?</p>	<p>Check the <a href="#">Supported Platforms</a> page for the version of Jira you are installing. This will give you info on supported operating systems, databases and browsers.</p> <p><b>Good to know:</b></p> <ul style="list-style-type: none"><li>• We don't support installing Jira on OSX or mac OS for production sites.</li><li>• The Jira installer includes Java (JRE) and Tomcat, so you don't need to install these separately.</li></ul>
<p>Do you want to run Jira as a service?</p>	<p>Running Jira as a service means that Jira will automatically start up when Linux is started.</p> <p><b>If you choose to run Jira as a service:</b></p> <ul style="list-style-type: none"><li>• You must use <code>sudo</code> to run the installer to be able to install Jira as a service.</li><li>• The installer will create a dedicated user account, <code>jira</code>, that will run the service.</li></ul> <p><b>If you choose not to run Jira as a service:</b></p> <ul style="list-style-type: none"><li>• You will start and stop Jira by running the <code>start-jira.sh</code> file in your Jira installation directory.</li><li>• Jira will be run as the user account that was used to install Jira, or you can choose to run as a dedicated user.</li><li>• Jira will need to be restarted manually if your server is restarted.</li></ul>

Is your database set up and ready to use?	<p>To run Jira in production you'll need an external database. Check the <a href="#">Supported platforms</a> page for the version you're installing for the list of databases we currently support. If you don't already have a database, PostgreSQL is free, easy to set up and has been extensively tested with Jira.</p> <p><b>Good to know:</b></p> <ul style="list-style-type: none"> <li>▪ Set up your database before you begin. Step-by-step guides for all supported databases are available in <a href="#">Connecting Jira applications to a database</a>.</li> <li>▪ Use UTF-8 character encoding.</li> <li>▪ If you're using Oracle or MySQL you'll need to download the driver for your database.</li> <li>▪ The embedded H2 database can be used for evaluating Jira, but you'll need to migrate to another database before running in production. You may find it easier to use external database from the start.</li> </ul>
Do you have a Jira license?	<p>You'll need a valid Jira Software Server, Jira Core Server or Jira Service Desk Server license to use JIRA.</p> <p><b>Good to know:</b></p> <ul style="list-style-type: none"> <li>• If you have not yet purchased a Jira application license you'll be able to create an evaluation license during setup.</li> <li>• If you already have a license key you'll be prompted to log in to <a href="https://my.atlassian.com">my.atlassian.com</a> to retrieve it, or you can enter the key manually during setup.</li> <li>• If you're migrating from Jira Cloud, you'll need a new license for Jira.</li> </ul>
Check some known issues	<p>For Linux installations, we've noticed some problems when displaying certain system text in the application (CAPTCHA and gadgets). Instead of showing regular alphanumeric letters, the text will appear to be garbled and look like symbols. To avoid this problem, you should install several fonts that are required by Jira. For more info, see <a href="#">Jira UI shows unreadable text</a>.</p>

## Install a Jira application

### 1. Download Jira

Download the installer for your operating system:

- Jira Core at <https://www.atlassian.com/software/jira/core/download>
- Jira Software at <https://www.atlassian.com/software/jira/download>
- Jira Service Management at <https://www.atlassian.com/software/jira/service-desk/download>

### 2. Run the installer

1. Make the installer executable.

Change to the directory where you downloaded Jira then execute this command:

<p><b>Jira Core</b></p>
<pre>\$ chmod a+x atlassian-jira-core-X.X.X-x64.bin</pre>
<p><b>Jira Software</b></p>
<pre>\$ chmod a+x atlassian-jira-software-X.X.X-x64.bin</pre>

**Jira Service Desk**

```
$ chmod a+x atlassian-servicedesk-X.X.X-x64.bin
```

Where `-X.X.X` is the Jira version you downloaded.

2. Run the installer, we recommend using `sudo` to run the installer as this will create a dedicated account to run Jira and allow you to run Jira as a service.

To use `sudo` to run the installer execute this command:

**Jira Core**

```
$ sudo ./atlassian-jira-core-X.X.X-x64.bin
```

**Jira Software**

```
$ sudo ./atlassian-jira-software-X.X.X-x64.bin
```

**Jira Service Desk**

```
$ sudo ./atlassian-servicedesk-X.X.X-x64.bin
```

Where `-X.X.X` is the Jira version you downloaded.

You can also choose to run the installer as with root user privileges.

3. Follow the prompts to install Jira. You'll be asked for the following info:
  - **Install type** choose option 2 (custom) for the most control.
  - **Destination directory** this is where Jira will be installed.
  - **Home directory** this is where Jira data like logs, search indexes and files will be stored.
  - **TCP ports** these are the HTTP connector port and control port Jira will run on. Stick with the default unless you're running another application on the same port.
  - **Install as service** this option is only available if you ran the installer as `sudo`.
4. Once installation is complete head to <http://localhost:8080> in your browser to begin the setup process. (Replace `8080` if you chose a different port during installation).

## Set up your Jira application

### 3. Choose set up method

Choose **I'll set it up myself**.

### 4. Connect to your database

1. If you've not already done so, it's time to create your database. See the 'Before you begin' section of this page for details.
2. Choose **My own database**.
3. Choose your database type then enter the details for your database.

JIRA connects to your database using a standard JDBC database connection. Connection pooling is handled within JIRA, you can change this using [JIRA configuration tool](#) later.

**If you're using Oracle or MySQL there's an extra step:**

- Download and extract the appropriate database JDBC drivers. See [Supported platforms](#) to get the right version.
- Drop the JAR file into your <jira-installation>/libfolder before continuing with the setup wizard.

#### In the setup wizard:

- **Driver Class Name** the Java class name for your database driver. If you're not sure, check the documentation for your database.
- **Database URL** the JDBC URL for your database. If you're not sure, check the documentation for your database.
- **Username and Password** A valid username and password that JIRA can use to access your database.

### 5. Set application properties

1. Give your Jirasite a name.
2. Choose whether your site should be private or anyone can sign up. You can change this later.
3. Enter your base URL - this is the address people will use to access your Jirasite.

### 5. Enter your license

Follow the prompts to log in to [my.atlassian.com](http://my.atlassian.com) to retrieve your license, or enter a license key.

### 6. Create your administrator account

Enter details for the administrator account. You can add more administrators after set up is complete.

### 7. Set up email notifications

Enter details of your mail server. This will allow Jira to send notifications when issues change.

### 8. Start using Jira

That's it! Your Jirasite is accessible from your base URL or a URL like this: `http://<computer_name_or_IP_address>:<port>`

Here's a few things that will help you get your team up and running:

- [Add and invite users](#) to get your team on board, or [configure user directories](#) for slightly bigger teams.
- [Create your first project](#) to have something to work on.
- [Configure SSL or HTTPS](#) to keep Jira and your team more secure.

## Troubleshooting

- Some anti-virus or other Internet security tools may interfere with the Jira installation process and prevent the process from completing successfully. If you experience or anticipate experiencing such an issue with your anti-virus/Internet security tool, disable this tool first before proceeding with the Jira installation.
- The [Linux OOM Killer](#) can sometimes kill Jira processes when memory on the server becomes too low. See [How to Configure the Linux Out-of-Memory Killer](#).

Head to [Installation Troubleshooting](#) for more help.

# Uninstalling Jira applications from Linux

This page describes the procedure for uninstalling Jira, which had been installed using the [Linux installer](#).

**i** If you wish to re-install Jira in 'unattended mode', do not uninstall your previous installation of Jira just yet. See [Using the silent installation feature](#) for more information.

## To uninstall Jira from Linux:

1. Open a Linux console.
2. Change directory (`cd`) to your Jira installation directory. For example:

```
cd /opt/atlassian/jira/
```

3. Execute the command `uninstall`

**i** This command must be executed as the same user account that was used to install Jira with the [Linux installer](#).

4. Follow the prompts to uninstall Jira from your computer.

## **i** Please note:

- All files within the Jira installation directory will be deleted (with the exception of the Tomcat log folder located in the Jira installation directory).
- The uninstaller will **NOT** delete:
  - The Jira database
  - The Jira home directory
  - Log files that were generated while Jira was running
- The uninstaller can be made to operate in unattended mode by specifying the `-q` option i.e. `uninstall -q`

# Installing Jira applications on Linux from Archive File

In this guide we'll run you through installing a Jira application in a production environment, with an external database, manually using a `tar.gz` file.

This method gives you the most control over the installation process.



## Other ways to install Jira:

- [Evaluation](#) - get your free trial up and running in no time.
- [Installer](#) - install Jira using the Linux installer.
- [Windows](#) - install Jira on a Windows server.

## On this page:

### Before you begin

#### Install a Jira application

1. Download Jira
2. Create the installation directory
3. Create the home directory
4. Check the ports
5. Start Jira

#### Set up your Jira application

6. Choose set up method
7. Connect to your database
8. Set application properties
9. Enter your license
10. Create your administrator account
11. Set up email notifications
12. Start using Jira

### Troubleshooting

## Before you begin

Before you install Jira, there are a few questions you need to answer.

Are you using a supported operating system and Java version?

Check the [Supported platforms](#) page for the version of Jira you are installing. This will give you info on supported operating systems, databases and browsers.

### Good to know:

- We don't support installing Jira on OS X or mac OS for production environments.
- You can use OpenJDK the JDK (Java Development Kit) or JRE (Java Runtime Environment).
- We only support the version of Apache Tomcat that is bundled with Jira.

<p>Are you using a 32-bit operating system?</p>	<p>If you're installing Jira on a 32-bit system, you need to decrease the maximum heap size available to Jira. The default for 64-bit systems is 2GB, which is too much for a 32-bit system, and may not fit into the available memory.</p> <p>Complete these steps after extracting files from the archive, but before starting Jira.</p> <p><b>Step 1: Rename the default setenv file.</b></p> <ol style="list-style-type: none"> <li>1. Go to <code>&lt;Jira-install-directory&gt;/bin</code>, and delete the <code>setenv.bat / .sh</code> file (or change its name).</li> <li>2. Rename <code>setenv32.bat / .sh</code> to <code>setenv.bat / .sh</code>. Jira will use this file on startup.</li> </ol> <p><b>Step 2: Add the properties to the jira-config.properties file.</b></p> <ol style="list-style-type: none"> <li>1. Go to Jira's home directory, and edit the <code>jira-config.properties</code> file. If the file isn't there, you can <a href="#">create it</a>.</li> <li>2. Add the following properties:</li> </ol> <pre style="border: 1px solid #ccc; padding: 5px;">jira.index.batch.maxrambuffermb=256 jira.index.interactive.maxrambuffermb=256</pre>
<p>Do you want to run Jira as a service?</p>	<p>Running Jira as a service means that your Jira application will automatically start up when your Linux server is started.</p> <p>You should use the <a href="#">Linux installer</a> if you want to run Jira as a service.</p> <p><b>If you choose not to run Jira as a service:</b></p> <ul style="list-style-type: none"> <li>• You will start your Jira application by running the <code>start-jira.sh</code> file in your Jira installation directory.</li> <li>• We recommend creating a dedicated user to run Jira. This user must have full read, write and execute access to the installation directory and home directory.</li> <li>• Jira will need to be restarted manually if your server is restarted.</li> </ul>
<p>What database do you plan to use?</p>	<p>To run Jira in production you'll need an external database. Check the <a href="#">Supported platforms</a> page for the version you're installing for the list of databases we currently support. If you don't already have a database, PostgreSQL is free, easy to set up and has been extensively tested with Jira.</p> <p><b>Good to know:</b></p> <ul style="list-style-type: none"> <li>■ Set up your database before you begin. Step-by-step guides for all supported databases are available in <a href="#">Connecting Jira applications to a database</a>.</li> <li>■ Use UTF-8 character encoding.</li> <li>■ If you're using Oracle or MySQL you'll need to download the driver for your database.</li> <li>■ The embedded H2 database can be used for evaluating Jira, but you'll need to migrate to another database before running in production. You may find it easier to use external database from the start.</li> </ul>

<p>Do you have a Jira license?</p>	<p>You'll need a valid JiraSoftware Server, JiraCore Server or JiraService Desk Server license to use JIRA.</p> <p><b>Good to know:</b></p> <ul style="list-style-type: none"> <li>• If you have not yet purchased a Jiraapplication license you'll be able to create an evaluation license during setup.</li> <li>• If you already have a license key you'll be prompted to log in to <a href="https://my.atlassian.com">my.atlassian.com</a> to retrieve it, or you can enter the key manually during setup.</li> <li>• If you're migrating from JiraCloud, you'll need a new licenseJira</li> </ul>
<p>Is your JAVA_HOME variable set correctly?</p>	<p>Before you install Jira, check that you're running a supported Java version and that the JAVA_HOME environment variable is set correctly.</p> <p><b>Jira applications can run with OpenJDK, Oracle JDK or JRE.</b></p> <p>To check your Java version:</p> <pre>\$ java -version</pre> <p>To check your JAVA_HOME variable is set correctly:</p> <pre>\$ echo \$JAVA_HOME</pre> <p>If you see a path to your Java installation directory, the JAVA_Home environment variable has been set correctly. If a path is not returned you'll need to set your JAVA_HOME environment variable manually before installing Jira.</p>
<p>Have you created a dedicated user to run Jira?</p>	<p>We strongly recommend running Jira as a dedicated user.</p> <p>You should create this user before you begin, so that when creating the installation and home directories, you can give this user appropriate read and write permissions.</p> <p>In this example, we'll create a user called jira:</p> <pre>\$ sudo /usr/sbin/useradd --create-home --comment "Account for running Jira Software" --shell /bin/bash jira</pre>

## Install a Jiraapplication

### 1.DownloadJira

Download the `tar.gz` file for your operating system:

- Jira Core at <https://www.atlassian.com/software/jira/core/download>
- Jira Software at <https://www.atlassian.com/software/jira/download>
- Jira Service Management at <https://www.atlassian.com/software/jira/service-desk/download>

### 2. Create the installation directory

1. Create your installation directory this is where Jirawill be installed. Avoid using spaces or special characters in the path. We'll refer to this directory as your `<installation-directory>`.

In this example we'll call our installation directory `jirasoftware`:

```
$ mkdir jirasoftware
```

2. Extract the Jira `tar.gz` file to your `<installation-directory>`. We recommend using a [GNU](#) version of the archive utility, especially on Solaris.

Change to the directory where you downloaded Jira then execute these commands:

```
$ tar -xzf atlassian-jira-software-X.X.X.tar.gz -C <installation-directory>
$ cd <installation-directory>
```

Replace `x.x.x` with your Jira version and `<installation-directory>` with the full path to the directory you created in the last step.

3. Give your dedicated Jira user read, write and execute permission to your `<installation-directory>`.

In this example we're changing ownership of the installation directory and giving the user `jira` read, write and execute permissions.

```
$ chown -R jira <installation-directory>
$ chmod -R u=rwx,go-rwx <installation-directory>
```

### 3. Create the home directory

1. Create your home directory this is where Jira application data like logs, search indexes and files will be stored. This should be separate to your installation directory, with no spaces or special characters in the path. Each Jira application needs its own home directory.

We'll refer to this directory as your `<home-directory>`.

In this example we'll call our home directory `jirasoftware-home`:

```
$ mkdir jirasoftware-home
```

2. Give your dedicated Jira user read, write and execute permissions to the `<home-directory>`.

In this example we're changing ownership of the home directory and giving the user `jira` read, write and execute permissions.

```
$ chown -R jira <home-directory>
$ chmod -R u=rwx,go-rwx <home-directory>
```

3. Tell Jira where to find your `<home-directory>` when it starts up. There are two ways to do this:

You can set an environment variable named `JIRA_HOME` in your operating system with the absolute path to your `<home-directory>`.

In Terminal, execute the following:

```
export JIRA_HOME=/path/to/home-directory
```

You can then specify the command above in a script used to start Jira.

Edit `<installation-directory>\atlassian-jira\WEB-INF\classes\jira-application.properties` file in any text editor.

After `jira.home` add the absolute path to your home directory (not a symlink), for example:

```
jira.home=/var/jirasoftware-home
```

#### 4. Check the ports

By default Jira listens on port 8080. If you have another application running on your server that uses the same ports, you'll need to tell Jira to use a different port.

To change the ports:

1. Edit `<installation-directory>\conf\server.xml`
2. Change the **Server** port (8005) and the **Connector** port (8080) to free ports on your server.

In the example below we've changed the **Server** port to 5005 and the **Connector** port to 5050.

```
<Server port="5005" shutdown="SHUTDOWN">
...
  <Service name="Catalina">
    <Connector port="5050"
      maxThreads="150"
      minSpareThreads="25"
      connectionTimeout="20000"
      enableLookups="false"
      maxHttpHeaderSize="8192"
      protocol="HTTP/1.1"
      useBodyEncodingForURI="true"
      redirectPort="8443"
      acceptCount="100"
      disableUploadTimeout="true"/>
```

If you are running on a Unix server and bind the ports below 1024 (such as port 80 for example), you will need to start Jira as root in order to successfully bind to the port.

#### 5. Start Jira

1. Run `<installation-directory>/bin/start-jira.sh` to start the setup process.

We recommend running Jira as your dedicated user.

```
$ su -u <user>
$ ./start-jira.sh
```

If you're using Ubuntu the command is a little different:

```
$ sudo su <user>
$ ./start-jira.sh
```

2. Go to <http://localhost:8080/> to launch Jira in your browser (change the port if you've updated the Connector port).

- Check your `JAVA_HOME` variable is set correctly.

Set up your Jira application

#### 6. Choose set up method

Choose **I'll set it up myself**.

## 7. Connect to your database

1. If you've not already done so, it's time to create your database. See the 'Before you begin' section of this page for details.
2. Choose **My own database**.
3. Choose your database type then enter the details for your database.

JIRA connects to your database using a standard JDBC database connection. Connection pooling is handled within JIRA, you can change this using [JIRA configuration tool](#) later.

### If you're using Oracle or MySQL there's an extra step:

- Download and extract the appropriate database JDBC drivers. See [Supported platforms](#) to get the right version.
- Drop the JAR file into your `<jira-installation>/lib` folder before continuing with the setup wizard.

### In the setup wizard:

- **Driver Class Name** the Java class name for your database driver. If you're not sure, check the documentation for your database.
- **Database URL** the JDBC URL for your database. If you're not sure, check the documentation for your database.
- **Username and Password** A valid username and password that JIRA can use to access your database.

## 8. Set application properties

1. Give your Jirasite a name.
2. Choose whether your site should be private or anyone can sign up. You can change this later.
3. Enter your base URL - this is the address people will use to access your Jirasite.

## 9. Enter your license

Follow the prompts to log in to [my.atlassian.com](http://my.atlassian.com) to retrieve your license, or enter a license key.

## 10. Create your administrator account

Enter details for the administrator account. You can add more administrators after set up is complete.

## 11. Set up email notifications

Enter details of your mail server. This will allow Jira to send notifications when issues change.

## 12. Start using Jira

That's it! Your Jirasite is accessible from your base URL or a URL like this: `http://<computer_name_or_IP_address>:<port>`

Here's a few things that will help you get your team up and running:

- [Add and invite users](#) to get your team on board, or [configure user directories](#) for slightly bigger teams.
- [Create your first project](#) to have something to work on.
- [Configure SSL or HTTPS](#) to keep Jira and your team more secure.

## Troubleshooting

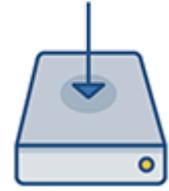
- Check your `JAVA_HOME` is set correctly.

- Use a [GNU](#) version of the unzip utility. There are known issues extracting the `tar.gz` file on Solaris and AIX. See '[extractBundledPlugins Couldn't find atlassian-bundled-plugins.zip on classpath](#)' [Due to Solaris TAR Utility](#).

Head to [Installation Troubleshooting](#) in our Knowledge Base for more help.

# Unattended installation

If you've previously installed Jira using the Windows or Linux installer, you can use a configuration file from your existing Jira installation (`response.varfile`) to re-install Jira in unattended mode, no user input required.



This can be useful when you have installed Jira on a test server and are ready to install on your production server with the same configuration.

## Good to know

- The `response.varfile` contains the options specified during the installation wizard steps of your previous Jira installation. Don't uninstall your previous Jira installation until after you've copied this file to your new install location.
- If you decide to modify the `response.varfile`, make sure all directory paths specified are absolute, for example, `sys.installationDir=C:\\Program Files\\Atlassian\\jira` (Windows) or `sys.installationDir=/opt/atlassian/jira` (Linux).

Unattended installations will fail if the file contains relative directory paths.

## Install a Jira application in unattended mode

1. Download the installer for your operating system:
  - Jira Core at <https://www.atlassian.com/software/jira/core/download>
  - Jira Software at <https://www.atlassian.com/software/jira/download>
  - Jira Service Management at <https://www.atlassian.com/software/jira/service-desk/download>
2. Copy `<installation-directory>/install4j/response.varfile` from your existing Jira installation to where you downloaded the installer.
3. In command prompt or terminal change directory (`cd`) to where you downloaded the installer.
4. Run the following command to install Jira:

### Windows

#### Jira Core

```
> atlassian-jira-core-X.X.X-x64.exe -q -varfile response.varfile
```

#### Jira Software

```
> atlassian-jira-software-X.X.X-x64.exe -q -varfile response.varfile
```

#### Jira Service Desk

```
> atlassian-servicedesk-X.X.X-x64.exe -q -varfile response.varfile
```

### Linux

#### Jira Core

```
$ atlassian-jira-core-X.X.X-x64.bin -q -varfile response.varfile
```

#### Jira Software

```
$ atlassian-jira-software-X.X.X-x64.bin -q -varfile response.varfile
```

**Jira Service Desk**

```
$ atlassian-servicedesk-X.X.X-x64.bin -q -varfile response.varfile
```

Where `-x.x.x` is the Jira version you downloaded.

`-q` instructs the installer to run in unattended mode (quietly). `-varfile` specifies the location and name of the configuration file containing the options used by the installer.

5. Jira will start automatically once the silent installation finishes.

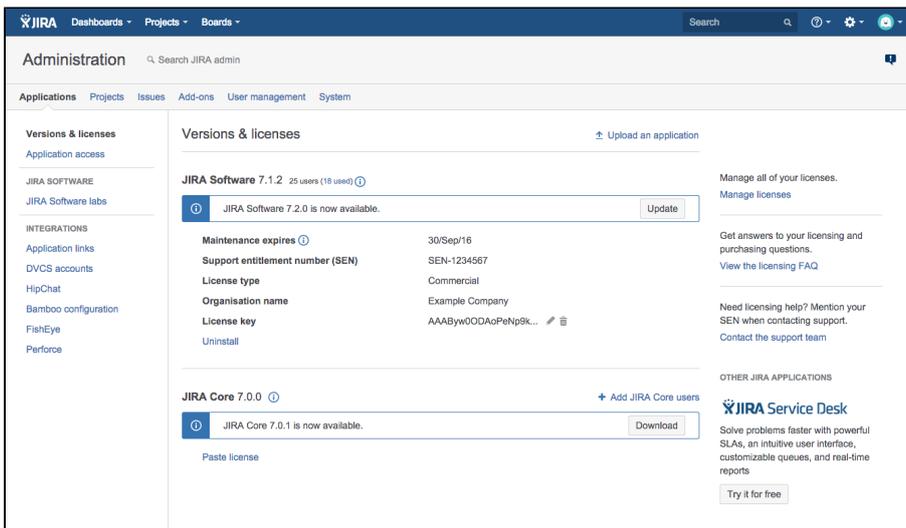
Finally, head to <http://localhost:<port>> to finish setting up Jira.

See the **Set up a Jira application** section on [Installing Jira applications on Windows](#) or [Installing Jira applications on Linux](#) for more info.

# Installing additional applications and version updates

After you have [installed your first Jira application](#) and have it running, you can install additional applications and update existing applications through the **Versions & licenses** page.

The image below shows you a typical installation of Jira Software. Note that only Jira Software is licensed. All updates for installed applications, in this instance Jira Core and Jira Software, are displayed. Jira Service Management is not installed, and you can see it's available to try on the right hand side.



## On this page:

- [Before you begin](#)
- [Discovering and installing additional applications](#)
- [Updating installed applications to the latest available update](#)
- [Updating installed applications to different version](#)
- [Updating Jira Core](#)
- [Options when you have no internet connection](#)

## Before you begin

- You need to have the **Jira Administrator** [global permission](#) to install additional applications or updates.
- To get a trial license for an application, you need your [Atlassian account](#) login details.
- If your Jira instance is not connected to the internet, you can download the additional application or required updates manually from the [Atlassian website](#). You can then get a trial license for your additional application at [my.atlassian.com](#). You may be asked to provide your [Server ID](#).

## Discovering and installing additional applications

If your Jira instance is connected to the internet, Jira will list additional applications available on the **Versions & licenses** page. You can download, install and license these applications directly on this page.

1. Select **Administration**  **Applications**. The list of additional application will display on the right-hand side of your **Versions & licensing** page.
2. Select the application you'd like to install, click on **Try it for free** and follow the prompts.
3. Your application is installed with a trial license and you're ready to go! For an overview of what additional functionality and features you'll be using, you can review the [applications and project types overview](#) page.

## Updating installed applications to the latest available update

1. Select **Administration**  **Applications**. Any applications that have updates available will display a message informing you of the latest available update.
2. Click the **Download** button in the message. A progress bar for your download will display, and confirm when it's completed.
3. Your application is up to date!

Jira Software and Jira Service Management can be updated in this way, while your server is running. Jira Core updates work differently and are described [below](#).

## Updating installed applications to different version

Sometimes you may need to update an application to a version that is not the latest available. This could be due to compatibility requirements with your JIRA Core version, or due to your license restricting you to updates prior to maintenance expiring.

When you want to update an installed application to a version that is not the latest available, you first need to download the version update file. You can browse available versions, along with their compatibility, on the Atlassian website:

- [Jira Software available versions](#)
- [Jira Service Management available versions](#)

Make sure the version you download is compatible with your JiraCore version. Once you've downloaded the update file, you can manually install it:

1. Select **Administration**  > **Applications**.
2. Select the **Upload an application** link.
3. Browse to the update file you downloaded.
4. Click the **Upload** button. A progress bar for your upload will display, and confirm when it's been uploaded and installed.
5. Your application is now updated to the version you selected.

## Updating JiraCore

The **Versions & licenses** page will notify you when an updated version of Jira Core is available. However, unlike version updates for Jira Software and Jira Service Management, updates for Jira Core cannot be applied while your server is running. Instead, **Versions & licenses** page will prompt you to download the installer for the new version.

If you want to download something other than the latest installer for JiraCore, you can download it from the Atlassian website:

- [Jira Core available versions](#)

To update your Jira Core installation, you should follow our [upgrade documentation](#), as this is an important step that requires planning and preparation.

## Options when you have no internet connection

If your Jira server is not connected directly to the Internet, or your firewall blocks connections to the [Atlassian Marketplace website](#), the **Versions & licenses** page will not be able to check for or apply version updates.

There are several scenarios that you may need to cover:

- To update your Jira Core installation, you should follow our [upgrade documentation](#), as this is an important step that requires planning and preparation.
- To update any other existing Jira applications, you can follow the steps set out in [Updating installed applications to a different version](#), making sure that the version you download is compatible with your JiraCore version.
- To install new applications, download the application file as described in [Updating installed applications to a different version](#). For this option, you'll also need to obtain a trial license for your additional application at [my.atlassian.com](#) so that you can update the license key manually after you've installed the application.

# Troubleshooting installation

If you run into issues with your Jira, refer to the information below to complete your installation.

## Installation tips

- If your web browser window shows an error the first time you try to access Jira, wait for 30 seconds or so and then refresh the page.
- If the command prompt window closes immediately, your JAVA\_HOME variable may not be set correctly.
- Some anti-virus or other Internet security tools may interfere with the Jira installation process and prevent the process from completing successfully. If you experience or anticipate experiencing such an issue with your anti-virus/Internet security tool, disable this tool first before proceeding with the Jira installation.
- If you install on Linux, the [Linux OOM Killer](#) can sometimes kill Jira processes when memory on the server becomes too low. See [How to Configure the Linux Out-of-Memory Killer](#).

## Known issues

Check this list of issues and solutions that's been compiled by our support: [Installation and upgrade troubleshooting](#).

## Resources

- Browse or search the [Jira Software documentation](#) site and the [Jira Software Server Knowledge Base](#). (Be sure to select the appropriate version in the dropdown.)
- Search our [Atlassian Community site](#) for articles and discussions pertaining to your topic. Or, ask a question to tap into Community experts both inside and outside of Atlassian.
- Create an issue at our [support site](#) so that our support engineers can assist you.
- If you purchased your license through an authorized Solution Partner, consult them to learn more about troubleshooting your instance.
- Looking for more assistance? We also offer Premier Support, which includes 24 x 7 support availability, health checks, dedicated senior support engineers, and more. Check out our [Premium Support offerings](#) for more information.

# Connecting Jira applications to a database

Jira requires a relational database to store its issue data. If you're setting up a completely new Jira installation, the [Jira setup wizard](#) will configure a database connection for you to either Jira's internal H2 or an external database.

Below you can find more detailed instructions for connecting Jira to a database:

- [Connecting Jira Data Center to Amazon Aurora](#)
- [Connecting Jira applications to Azure SQL](#)
- [Connecting Jira applications to PostgreSQL](#)
- [Connecting Jira applications to MySQL 5.7](#)
- [Connecting Jira applications to MySQL 8.0](#)
- [Connecting Jira applications to Oracle](#)
- [Connecting Jira applications to SQL Server 2016](#)
- [Connecting Jira applications to SQL Server 2017](#)
- [Tuning database connections](#)
- [Encrypting database password](#)
- [Switching databases](#)
- [Connecting Jira applications to SQL Server 2019](#)

## Which database?

Your choice of database can significantly affect your subsequent experience of Jira administration. If you have a choice of databases, first read [our list of supported databases](#).

 If you are looking for a low-cost solution, consider using [MySQL](#) or [PostgreSQL](#), as both of these are open source (free) software.

## Upgrading Jira or migrating to another server?

If you are [upgrading Jira manually](#) or [migrating Jira to another server](#), and do not have access to a pre-existing `dbconfig.xml` file, you will need to re-configure your database connection. This results in `adbconfig.xml` file (being created in the [Jira home directory](#) of your new Jira installation), whose content defines your Jira database connection.

You can re-configure your database connection with either the [Jira configuration tool](#), or manually by editing the `dbconfig.xml` file. You can find the details on how to do it in the specific instructions for each database, listed above.

## Data migration

To transfer your issue data from one database to another, please refer to the instructions for [Switching databases](#).

# Connecting Jira Data Center to Amazon Aurora

These instructions will help you connect Jira to an existing Amazon Aurora PostgreSQL database.

## Amazon Aurora is only supported on a Data Center license

Jira Data Center supports the use of a single-writer, PostgreSQL-compatible Amazon Aurora clustered database. A typical production-grade cluster includes one or more readers in a different availability zone. If the writer fails, Amazon Aurora will automatically promote one of the readers to take its place. For more information, see [Amazon Aurora Features: PostgreSQL-Compatible Edition](#).

## On this page:

- [Before you begin](#)
- [1. Create and configure the PostgreSQL database](#)
- [2. Configure your Jira server to connect to your PostgreSQL database](#)
- [3. Start Jira](#)

## Before you begin

- Check whether your version of Amazon Aurora is supported. See [Supported platforms](#).
- Shut down Jira before you begin, unless you are running the setup wizard.

## 1. Create and configure the PostgreSQL database

### Use the Quick Start for new deployments

If you are deploying a new Jira Data Center with Amazon Aurora from scratch, we recommend that you use the [AWS Quick Start for Jira](#). This Quick Start lets you configure a PostgreSQL-compatible Amazon Aurora cluster with one writer and two readers (preferably in separate availability zones). See [Getting started with Jira Data Center in AWS](#) for more information.

Jira Data Center specifically supports the use of an Amazon Aurora cluster with the following configuration:

- It must have only one writer, replicating to one or more readers.
- Your PostgreSQL engine must be version 9.6 or higher.

See [Supported platforms](#) for more details.

## AWS documentation

AWS has some helpful guides for setting up an Aurora database and migrating to it:

- [Modular Architecture for Amazon Aurora PostgreSQL](#): a Quick Start that guides you through the deployment of a PostgreSQL-compatible Aurora Database cluster. This cluster is similar to the one set up by the [AWS Quick Start for Jira](#) (one writer and two readers, preferably in different availability zones).
- [Upgrading the PostgreSQL DB Engine for Amazon RDS](#): shows you how to upgrade your database engine to a supported version before migrating it to Amazon Aurora.
- [Migrating Data to Amazon Aurora PostgreSQL](#): contains instructions for migrating from Amazon RDS to a PostgreSQL-compatible Amazon Aurora cluster.
- [Best Practices with Amazon Aurora PostgreSQL](#): contains additional information about best practices and options for migrating data to a PostgreSQL-compatible Amazon Aurora cluster.

Amazon also offers an [AWS Database Migration Service](#) to facilitate a managed migration. This service offers minimal downtime, and supports migrations to Aurora from a wide variety of source databases.

## 2. Configure your Jira server to connect to your PostgreSQL database

There are two ways to configure your Jira server to connect to your Amazon Aurora database:

- **Using the Jira setup wizard** Use this method if you have just installed Jira, and you are setting it up for the first time. Your settings will be saved to the `dbconfig.xml` file in your [Jira home directory](#).
- **Using the Jira configuration tool** Use this method if you have an existing Jira instance. Your settings will be saved to the `dbconfig.xml` file in your [Jira home directory](#).

### Instructions for each configuration method

#### Jira setup wizard

The [Jira setup wizard](#) will display when you access Jira for the first time in your browser.

1. In the first screen, 'Configure Language and Database', set **Database Connection to My own database**.
2. Set **Database Type to Aurora PostgreSQL 9.6 (DC Only)**.
3. Fill out the fields, as described in the [Database connection fields](#) section below.
4. Test your connection and save.

#### Jira configuration tool

1. Run the Jira configuration tool as follows:
  - **Windows:** Open a command prompt and run `config.bat` in the `bin` sub-directory of the [Jira installation directory](#).
  - **Linux/Unix:** Open a console and execute `config.sh` in the `bin` sub-directory of the [Jira installation directory](#).

 This command might fail with the error as described in [Unable to Start Jira applications Config Tool due to No X11 DISPLAY variable was set error](#). If it happens, refer to this article for the workaround.

2. Navigate to the **Database tab**
3. Set **Database Type to Aurora PostgreSQL 9.6 (DC Only)**.
4. Fill out the fields, as described in the [Database connection fields](#) section below.
5. Test your connection and save.
6. Restart Jira.

### Database connection fields

Setup Wizard / Configuration Tool	dbconfig.xml tag	Description
<b>Hostname</b>	Located in the <code>&lt;url&gt;</code> tag (bold text in example below): <code>&lt;url&gt;jdbc:postgresql://<b>dbserver</b>:5432/jiradb?targetServerType=master&lt;/url&gt;</code>	The name or IP address of the machine that the PostgreSQL server is installed on.
<b>Port</b>	Located in the <code>&lt;url&gt;</code> tag (bold text in example below): <code>&lt;url&gt;jdbc:postgresql://dbserver:<b>5432</b>/jiradb?targetServerType=master&lt;/url&gt;</code>	The TCP/IP port that the PostgreSQL server is listening on. You can leave this blank to use the default port.
<b>Database</b>	Located in the <code>&lt;url&gt;</code> tag (bold text in example below): <code>&lt;url&gt;jdbc:postgresql://dbserver:</code>	The name of your PostgreSQL database (into which Jira will save its data). You should have created this in <a href="#">Step 1</a> above.

	<code>5432/jiradb?targetServerType=master&lt;/url&gt;</code>	
<b>Username</b>	Located in the <code>&lt;username&gt;</code> tag (see bold text in example below): <code>&lt;username&gt;jiradbuser&lt;/username&gt;</code>	The user that JIRA uses to connect to the PostgreSQL server. You should have created this in <a href="#">Step 1</a> above.
<b>Password</b>	Located in the <code>&lt;password&gt;</code> tag (see bold text in example below): <code>&lt;password&gt;jiradbuser&lt;/password&gt;</code>	The user's password used to authenticate with the PostgreSQL server.
<b>Schema</b>	Located in the <code>&lt;schema-name&gt;</code> tag (see bold text in example below): <code>&lt;schema-name&gt;public&lt;/schema-name&gt;</code>	The name of the schema that your PostgreSQL database uses.

### Schema requirements

PostgreSQL 7.2 and later require a schema to be specified in the `<schema-name/>` element. If your PostgreSQL database uses the default 'public' schema, this should be specified in the `<schema-name/>` element as shown below. Ensure that your database schema name is lower-case, as JIRA cannot work with PostgreSQL databases whose schema names contain upper-case characters.

### Sample dbconfig.xml file

For more information about the child elements of `<jdbc-datasource/>` beginning with `pool` in the `dbconfig.xml` file below, see [Tuning database connections](#).

```
<?xml version="1.0" encoding="UTF-8"?>

<jira-database-config>
  <name>defaultDS</name>
  <delegator-name>default</delegator-name>
  <database-type>postgres72</database-type>
  <schema-name>public</schema-name>
  <jdbc-datasource>
    <url>jdbc:postgresql://dbserver:5432/jiradb?targetServerType=master</url>
    <driver-class>org.postgresql.Driver</driver-class>
    <username>jiradbuser</username>
    <password>password</password>
    <pool-min-size>20</pool-min-size>
    <pool-max-size>20</pool-max-size>
    <pool-max-wait>30000</pool-max-wait>
    <pool-max-idle>20</pool-max-idle>
    <pool-remove-abandoned>true</pool-remove-abandoned>
    <pool-remove-abandoned-timeout>300</pool-remove-abandoned-timeout>

    <validation-query>select version();</validation-query>
    <min-evictable-idle-time-millis>60000</min-evictable-idle-time-millis>
    <time-between-eviction-runs-millis>300000</time-between-eviction-runs-millis>

    <pool-test-on-borrow>false</pool-test-on-borrow>
    <pool-test-while-idle>true</pool-test-while-idle>

  </jdbc-datasource>
</jira-database-config>
```

## 3. Start Jira

You should now have Jira Data Center configured to connect to your Amazon Aurora database. The next step is to start it up!

✔ **Congratulations, you now have Jira Data Center connected to your PostgreSQL database.**

# Connecting Jira applications to Azure SQL

These instructions will help you connect Jira Server or Jira Data Center to an Azure SQL database.

Before you begin

- If you're [Migrating Jira applications to another server](#), create an export of your data as an [XML backup](#). You will then be able to transfer data from your old database to your new database, as described in [Switching databases](#).
- If you're not too familiar with Azure, you can use our official templates to easily deploy Jira Data Center into Azure.

## 1. Create an Azure SQL database

Create an Azure SQL database. See [Quickstart: Create a single database in Azure](#).

### Requirements

- **Collation:** When creating the database, make sure to set the right collation in **Additional settings**, as you won't be able to change it later. Collation types supported by Jira are `SQL_Latin1_General_CP437_CI_AI` and `Latin1_General_CI_AI`.

## 2. Allow Jira to connect to the database

You need to add the IP address of your Jira server to the database's firewall rules to allow Jira to connect to your Azure SQL database. See [Azure SQL database firewall rules](#).

## 3. Configure Jira to connect to the database

There are two ways to configure your Jira server to connect to your Azure SQL database.



### Finding connection strings

When connecting Jira to the database, you'll need to provide connection details, such as hostname, port number, and database name. You can find them in the Azure portal by opening your deployed database and going to **Connection strings**.

- **Using the Jira setup wizard** Use this method, if you have just installed Jira and are setting it up for the first time. Your settings will be saved to the `dbconfig.xml` file in your [Jira application home directory](#).

The [Jira setup wizard](#) will display when you access Jira for the first time in your browser.

1. In the first screen, 'Configure Language and Database', set **Database Connection** to **My own database**.
  2. Set **Database Type** to **Microsoft SQL Server**.
  3. Fill out the fields, as described in the [Database connection fields](#) section below.
  4. Test your connection and save.
- **Using the Jira configuration tool** Use this method, if you have an existing Jira instance. Your settings will be saved to the `dbconfig.xml` file in your [Jira application home directory](#).
    1. Run the Jira configuration tool as follows:
      - **Windows:** Open a command prompt and run `runconfig.bat` in the `bin\sub`-directory of the [Jira installation directory](#).
      - **Linux/Unix:** Open a console and execute `config.sh` in the `bin\sub`-directory of the [Jira installation directory](#).
    2. Navigate to the **Database** tab and set **Database type** to **Microsoft SQL Server**.
    3. Fill out the fields, as described in the [Database connection fields](#) section below.

4. Test your connection and save.
5. Restart Jira.

### Database connection fields

The sections below describe the fields you'll need to fill out when connecting Jira to your database.

Field	Description / Example
<b>Hostname</b>	The name or IP address of the Azure SQL server. <b>Example:</b> <code>sqlserver.database.windows.net</code>
<b>Port</b>	The TCP/IP port that the Azure SQL server is listening on. You can leave this blank to use the default port. <b>Default:</b> 1433
<b>Database</b>	The name of your Azure SQL database (into which Jira will save its data). <b>Example:</b> <code>jiradb</code>
<b>Username</b>	The user that Jira uses to connect to the SQL Server server. <b>Example:</b> <code>jiradbuser@sqlserver</code>
<b>Password</b>	The user's password used to authenticate with the Azure SQL server.
<b>Schema</b>	The name of the schema that your Azure SQL database uses. <b>Default:</b> <code>dbo</code>

Field	Description / Example
<b>Hostname</b>	The name or IP address of the Azure SQL server. - <code>&lt;url&gt;jdbc:sqlserver://;serverName=<b>sqlserver.database.windows.net</b>; portNumber=1433;databaseName=jiradb&lt;/url&gt;</code>
<b>Port</b>	The TCP/IP port that the Azure SQL server is listening on. You can leave this blank to use the default port. - <code>&lt;url&gt;jdbc:sqlserver://;serverName=sqlserver.database.windows.net; portNumber=<b>1433</b>;databaseName=jiradb&lt;/url&gt;</code>
<b>Database</b>	The name of your Azure SQL database (into which Jira will save its data). - <code>&lt;url&gt;jdbc:sqlserver://;serverName=sqlserver.database.windows.net; portNumber=1433;databaseName=<b>jiradb</b>&lt;/url&gt;</code>

<b>Username</b>	The user that Jira uses to connect to the SQL Server server. - <username> <b>jiradbuser</b> </username>
<b>Password</b>	The user's password used to authenticate with the Azure SQL server. - <password> <b>yourpassword</b> </password>
<b>Schema</b>	The name of the schema that your Azure SQL database uses. - <schema-name> <b>dbo</b> </schema-name>

### Sample dbconfig.xml file

For more information about the child elements of <jdbc-datasource/> beginning with pool in the dbconfig.xml file above, see [Tuning database connections](#).

```
<jira-database-config>
  <name>defaultDS</name>
  <delegator-name>default</delegator-name>
  <database-type>mssql</database-type>
  <schema-name>dbo</schema-name>
  <jdbc-datasource>
    <url>jdbc:sqlserver://;serverName=sqlserver.database.windows.net;portNumber=1433;
databaseName=jiradb</url>
    <driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>
    <username>adminsqli@sqlserver</username>
    <password>T3ddybear</password>
    <pool-min-size>20</pool-min-size>
    <pool-max-size>20</pool-max-size>
    <pool-max-wait>30000</pool-max-wait>
    <validation-query>select 1</validation-query>
    <min-evictable-idle-time-millis>60000</min-evictable-idle-time-millis>
    <time-between-eviction-runs-millis>300000</time-between-eviction-runs-millis>
    <pool-max-idle>20</pool-max-idle>
    <pool-remove-abandoned>true</pool-remove-abandoned>
    <pool-remove-abandoned-timeout>300</pool-remove-abandoned-timeout>
    <pool-test-on-borrow>false</pool-test-on-borrow>
    <pool-test-while-idle>true</pool-test-while-idle>
  </jdbc-datasource>
</jira-database-config>
```

## 4. Schedule regular database maintenance tasks

To achieve and maintain optimal MS SQL performance, schedule daily maintenance tasks to update database statistics.

### Schedule a daily maintenance task for hot tables

Hot tables are the most active tables in your database. For example, `propertyentry`, `changeitem`, and `changegroup` are large data tables that are used frequently and require regular updating of statistics.

To set up a daily maintenance task for hot tables, run the following command:

```
UPDATE STATISTICS <table.name>
```

### Schedule a weekly maintenance task for the whole database

To set up a weekly maintenance task for the whole database, run the following command:

```
UPDATE STATISTICS with fullscan
```

 For large databases, updating statistics with `fullscan` might take a long time to complete. To minimize the impact on your production environment, schedule this maintenance task for off-peak hours.

For more information on how to update database statistics, see the [official Microsoft documentation](#).

# Connecting Jira applications to PostgreSQL

These instructions will help you connect Jira to a [PostgreSQL](#) database.

## Before you begin

- Check whether your version of PostgreSQL is supported. See [Supported platforms](#).
- If you are [migrating Jira to another server](#), create an export of your data as an [backup](#). You will then be able to transfer data from your old database to your new database, as described in [Switching databases](#).
- Shut down Jira before you begin, unless you are running the setup wizard.

### On this page:

- [Before you begin](#)
- [1. Create and configure the PostgreSQL database](#)
- [2. Configure your Jira server to connect to your PostgreSQL database](#)
- [3. Start Jira](#)

## 1. Create and configure the PostgreSQL database

### Accept remote TCP connections (remote PostgreSQL server only)

If you are connecting Jira to a remote PostgreSQL server (i.e. if your PostgreSQL server is not installed locally on your Jira server host system), you will need to configure your `data/postgresql.conf` and `data/pg_hba.conf` files to accept remote TCP connections from your Jira server's IP address.

The following PostgreSQL documentation contains information on the appropriate `listen_addresses` value in the `postgresql.conf` file as well as the `pg_hba.conf` file:

- [PostgreSQL 11 documentation Connections and Authentication](#)
- [PostgreSQL 10 documentation Connections and Authentication](#)
- [PostgreSQL 9.6 documentation Connections and Authentication](#)

After you modify the `data/postgresql.conf` and `data/pg_hba.conf` files, restart PostgreSQL for the changes to take effect.

### Create users and databases for your version of PostgreSQL

You can find information on creating users and databases for your version of PostgreSQL on their [website](#).

1. Create a database user (login role) which Jira will connect as (e.g. `jradbuser`).  
**Remember this database user name**, as it will be used to configure Jira's connection to this database in subsequent steps.
2. Create a database for Jira to store issues in (e.g. `jradb`) with Unicode collation.  
**Remember this database name**, as it will be used to configure Jira's connection to this database in subsequent steps.

```
CREATE DATABASE jradb WITH ENCODING 'UNICODE' LC_COLLATE 'C' LC_CTYPE 'C' TEMPLATE template0;
```

Or from the command-line:

```
$ createdb -E UNICODE -l C -T template0 jradb
```

3. Ensure that the user has permissions to connect to the database, and to create and write to tables in the database.

```
GRANT ALL PRIVILEGES ON DATABASE <Database Name> TO <Role Name>
```

- To verify that the privileges were granted successfully, connect to the database and run the `\z` command.

**i** To achieve and maintain optimal PostgreSQL performance, schedule daily maintenance tasks to update database statistics.

For information on how to set up regular maintenance tasks, see [Optimize and Improve PostgreSQL Performance with VACUUM, ANALYZE, and REINDEX](#)

## 2. Configure your Jiraserver to connect to your PostgreSQL database

There are two ways to configure your Jiraserver to connect to your PostgreSQL database:

- **Using the Jirasetup wizard** Use this method if you have just installed Jira, and you are setting it up for the first time. Your settings will be saved to the `dbconfig.xml` file in your [Jira home directory](#).
- **Using the Jiraconfiguration tool** Use this method if you have an existing Jira instance. Your settings will be saved to the `dbconfig.xml` file in your [Jira home directory](#).

### Instructions for each configuration method

#### Jirasetup wizard

The [Jira setup wizard](#) will display when you access Jira for the first time in your browser.

- In the first screen, 'Configure Language and Database', set **Database Connection** to **My own database**.
- Set **Database Type** to **PostgreSQL**.
- Fill out the fields, as described in the [Database connection fields](#) section below.
- Test your connection and save.

#### Jiraconfiguration tool

- Run the Jiraconfiguration tool as follows:
  - **Windows:** Open a command prompt and run `config.bat` in the `bin` sub-directory of the [Jira installation directory](#).
  - **Linux/Unix:** Open a console and execute `config.sh` in the `bin` sub-directory of the [Jira installation directory](#).

**i** This command might fail with the error as described in [Unable to Start Jira applications Config Tool due to No X11 DISPLAY variable was set error](#). If it happens, refer to this article for the workaround.

- Navigate to the **Database** tab, and set **Database type** to **PostgreSQL**.
- Fill out the fields, as described in the [Database connection fields](#) section below.
- Test your connection and save.
- Restart Jira.

### Database connection fields

Setup Wizard / Configuration Tool	dbconfig.xml	Description

<b>Hostname</b>	Located in the <code>&lt;url&gt;</code> tag (bold text in example below): <code>&lt;url&gt;jdbc:postgresql://<b>dbserver</b>:5432/jiradb&lt;/url&gt;</code>	The name or IP address of the machine that the PostgreSQL server is installed on.
<b>Port</b>	Located in the <code>&lt;url&gt;</code> tag (bold text in example below): <code>&lt;url&gt;jdbc:postgresql://dbserver:<b>5432</b>/jiradb&lt;/url&gt;</code>	The TCP/IP port that the PostgreSQL server is listening on. You can leave this blank to use the default port.
<b>Database</b>	Located in the <code>&lt;url&gt;</code> tag (bold text in example below): <code>&lt;url&gt;jdbc:postgresql://dbserver:5432/<b>jiradb</b>&lt;/url&gt;</code>	The name of your PostgreSQL database (into which Jira will save its data). You should have created this in <a href="#">Step 1</a> above.
<b>Username</b>	Located in the <code>&lt;username&gt;</code> tag (see bold text in example below): <code>&lt;username&gt;<b>jiradbu</b>ser&lt;/username&gt;</code>	The user that JIRA uses to connect to the PostgreSQL server. You should have created this in <a href="#">Step 1</a> above.

<b>Password</b>	Located in the <code>&lt;password&gt;</code> tag (see bold text in example below): <code>&lt;password&gt;<b>jiradbuser</b>&lt;/password&gt;</code>	The user's password used to authenticate with the PostgreSQL server.
<b>Schema</b>	Located in the <code>&lt;schema-name&gt;</code> tag (see bold text in example below): <code>&lt;schema-name&gt;<b>public</b>&lt;/schema-name&gt;</code>	<p>The name of the schema that your PostgreSQL database uses.</p> <p>PostgreSQL 7.2 and later require a schema to be specified in the <code>&lt;schema-name/&gt;</code> element. If your PostgreSQL database uses the default 'public' schema, this should be specified in the <code>&lt;schema-name/&gt;</code> element as shown below. Ensure that your database schema name is lower-case, as JIRA cannot work with PostgreSQL databases whose schema names contain upper-case characters.</p> <div data-bbox="555 824 1433 925" style="border: 1px solid #ccc; padding: 5px;"> <p> We recommend using public schema as a custom one might cause issues. See <a href="#">JRASERVER-64886</a></p> </div>

### Sample dbconfig.xml file

For more information about the child elements of `<jdbc-datasource/>` beginning with `pool` in the `dbconfig.xml` file above, see [Tuning database connections](#).

```
<?xml version="1.0" encoding="UTF-8"?>

<jira-database-config>
  <name>defaultDS</name>
  <delegator-name>default</delegator-name>
  <database-type>postgres72</database-type>
  <schema-name>public</schema-name>
  <jdbc-datasource>
    <url>jdbc:postgresql://dbserver:5432/jiradb</url>
    <driver-class>org.postgresql.Driver</driver-class>
    <username>jiradbuser</username>
    <password>password</password>
    <pool-min-size>20</pool-min-size>
    <pool-max-size>20</pool-max-size>
    <pool-max-wait>30000</pool-max-wait>
    <pool-max-idle>20</pool-max-idle>
    <pool-remove-abandoned>true</pool-remove-abandoned>
    <pool-remove-abandoned-timeout>300</pool-remove-abandoned-timeout>

    <validation-query>select version();</validation-query>
    <min-evictable-idle-time-millis>60000</min-evictable-idle-time-millis>
    <time-between-eviction-runs-millis>300000</time-between-eviction-runs-millis>

    <pool-test-on-borrow>false</pool-test-on-borrow>
    <pool-test-while-idle>true</pool-test-while-idle>

  </jdbc-datasource>
</jira-database-config>
```

## 3. Start Jira

You should now have Jira configured to connect to your PostgreSQL database. The next step is to start it up!

✔ **Congratulations, you now have Jira connected to your PostgreSQL database.**

# Connecting Jira applications to MySQL 5.7

These instructions will help you connect Jira to a supported MySQL database.

- [Before you begin](#)
- [Create and configure the MySQL database](#)
- [Copy the MySQL JDBC driver](#)
- [Configure Jira to connect to the database](#)
- [Start Jira](#)
- [Database connection fields](#)
- [Known issues](#)

## Before you begin

Here is some prerequisite information you should know about:

- Check [known issues](#).
- If you are [migrating Jira to another server](#), create an export of your data as an [XML backup](#). You will then be able to transfer data from your old database to your new database, as described in [Switching databases](#).
- If you plan to set up Confluence and Jira on the same MySQL server, read the [Confluence MySQL setup guide](#). Confluence requirements are more strict than Jira's, so you should configure MySQL to suit Confluence. This configuration will work for Jira, too.
- Shut down Jira before you begin, unless you are running the setup wizard.

## 1. Create and configure the MySQL database

 When creating the database, remember your **database name**, **user name**, and **port number**, because you'll need them later to connect Jira to your database.

1. Create a database user which Jira will connect as, for example **jiradbuser**.
2. Create a database for Jira to store issues in, for example **jiradb**.

The database must have a character set of UTF8. To set it, enter the following command from within the MySQL command client:

```
CREATE DATABASE jiradb CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
```

3. Make sure the user has permission to connect to the database, and permission to create and populate tables. You can provide these permissions with the following commands.

### MySQL 5.7.0 - 5.7.5:

```
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER,INDEX on <JIRADB>.* TO  
'<USERNAME>'@'<JIRA_SERVER_HOSTNAME>' IDENTIFIED BY '<PASSWORD>';  
flush privileges;
```

### MySQL 5.7.6 and later (must also include the REFERENCES permission):

```
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,REFERENCES,ALTER,INDEX on <JIRADB>.* TO
'<USERNAME>'@'<JIRA_SERVER_HOSTNAME>' IDENTIFIED BY '<PASSWORD>';
flush privileges;
```

4. Edit the `my.cnf` or `my.ini` (Windows) file in your MySQL Server (for detailed instructions on editing these files, see [MySQL Option Files](#)).
5. Locate the `[mysqld]` section in the file, and add or modify the following parameters:

- Set the default storage engine to InnoDB:

```
[mysqld]
...
default-storage-engine=INNODB
...
```

- Specify the character set used by the database server:

```
[mysqld]
...
character_set_server=utf8mb4
...
```

- Set the default row format to `DYNAMIC`:

```
[mysqld]
...
innodb_default_row_format=DYNAMIC
...
```

- Enable the large prefix:

```
[mysqld]
...
innodb_large_prefix=ON
...
```

- Set the InnoDB file format to Barracuda:

```
[mysqld]
...
innodb_file_format=Barracuda
...
```

- Specify the value of `innodb_log_file_size` to be at least 2G:

```
[mysqld]
...
innodb_log_file_size=2G
...
```

- Ensure the `sql_mode` parameter does not specify `NO_AUTO_VALUE_ON_ZERO`

```
// remove this if it exists
sql_mode = NO_AUTO_VALUE_ON_ZERO
```

6. Restart your MySQL server for the changes to take effect.

Use the Windows Services manager to restart the service.

Run one of the following commands, depending on your setup:

```
/etc/init.d/mysqld stop
```

```
/etc/init.d/mysql stop
```

```
service mysqld stop
```

Then, run the same command, replacing `stop` with `start`.

## 2. Copy the MySQL JDBC driver

Copy the MySQL JDBC driver to the Jira installation directory.

1. Download one of the recommended MySQL drivers:
  - [MySQL Connector/J 8.0 driver](#)
  - [MySQL Connector/J 5.1 driver](#)
2. Copy the driver to the following directory:

```
<Jira-installation-directory>/lib
```

 If you are installing Jira using the Windows installer, you will need to do this step after running the Windows installer, but **before** running the setup wizard.

3. Restart the Jira service.
4. If you are installing Jira, skip the rest of the instructions on this page and access Jira in your browser to run the setup wizard instead.

## 3. Configure Jira to connect to the database

There are two ways to configure your Jira server to connect to your MySQL database:

### Setup wizard

Use the setup wizard if you have just installed Jira, and are setting it up for the first time. Your settings will be saved to the `dbconfig.xml` file in your [Jira home directory](#).

The [Jira setup wizard](#) will display when you access Jira for the first time in your browser.

1. In the first screen, 'Configure Language and Database', set **Database Connection** to **My own database**.
2. Set **Database Type** to **MySQL 5.7+**.
3. Fill out the fields, as described in the [Database connection fields](#) section below.
4. Test your connection and save.

### Configuration tool

Use the configuration tool if you have an existing Jira instance. Your settings will be saved to the `dbconfig.xml` file in your [Jira home directory](#).

1. Run the Jira configuration tool as follows:
  - **Windows:** Open a command prompt and run `config.bat` in the `bin` sub-directory of the [Jira installation directory](#).

- **Linux/Unix:** Open a console and execute `config.sh` in the `bin` sub-directory of the [Jira installation directory](#).

**i** This command might fail with the error as described in [Unable to Start Jira applications Config Tool due to No X11 DISPLAY variable was set error](#). If it happens, refer to this article for the workaround.

2. Navigate to the **Databasetab** and set **Database type** to **MySQL 5.7+**.
3. Fill out the fields, as described in the [Database connection fields](#) section below.
4. Test your connection and save.
5. Restart Jira.

## 4. Start Jira

You should now have Jira configured to connect to your MySQL database. The next step is to start it up!

### Database connection fields

The table below explains database connection fields that you can find in the setup wizard, Jira configuration tool, or the `dbconfig.xml` file.

Setup wizard / Configuration tool	Description	dbconfig.xml
<b>Hostname</b>	The name or IP address of the machine that the MySQL server is installed on.	<p>Located in the <code>&lt;url&gt;</code> tag. In the example below, <b>dbserver</b>.</p> <pre>&lt;url&gt;jdbc:mysql://dbserver:3306/jiradb? useUnicode=true&amp; characterEncoding=UTF8&amp; sessionVariables=default_storage_engine=InnoDB &lt;/url&gt;</pre> <p>If you use an IPv6 address, the URL needs to look like this:</p> <pre>&lt;url&gt;jdbc:mysql://address=(protocol=tcp) (host=dbserver)(port=3306)/jiradb? useUnicode=true&amp; characterEncoding=UTF8&amp; sessionVariables=default_storage_engine=InnoDB &lt;/url&gt;</pre>
<b>Port</b>	The TCP/IP port that the MySQL server is listening on. You can leave this blank to use the default port.	<p>Located in the <code>&lt;url&gt;</code> tag. In the example below, <b>3306</b>.</p> <pre>&lt;url&gt;jdbc:mysql://dbserver:3306/jiradb? useUnicode=true&amp; characterEncoding=UTF8&amp; sessionVariables=default_storage_engine=InnoDB &lt;/url&gt;</pre>

<b>Database</b>	The name of your MySQL database (into which Jira will save its data). You should have created this in <a href="#">Step 1</a> above.	Located in the <code>&lt;url&gt;</code> tag. In the example below, <b>jiradb</b> .  <pre>&lt;url&gt;jdbc:mysql://dbserver:3306/jiradb ? useUnicode=true&amp; characterEncoding=UTF8&amp; sessionVariables=default_storage_engine=InnoDB &lt;/url&gt;</pre>
<b>Username</b>	The user that Jira uses to connect to the MySQL server. You should have created this in <a href="#">Step 1</a> above.	<pre>&lt;username&gt;jiradbuser&lt;/username&gt;</pre>
<b>Password</b>	The user's password used to authenticate with the MySQL server.	<pre>&lt;password&gt;jiradbuser&lt;/password&gt;</pre>

- For more information about the child elements of `<jdbc-datasource/>` beginning with `pool` in the `dbconfig.xml` file above, see [Tuning database connections](#).
- Both the Jira setup wizard and database configuration tool also add the element `<validation-query>select 1</validation-query>` to this file, which is usually required when running Jira with default MySQL installations. See [Surviving connection closures](#) for more information.
- The database URL in the example below assumes a UTF-8 database i.e. that your database was created using a command similar to `create database jiradb character set utf8`; If you do not specify `character set utf8` when creating this database, you risk getting 'Data truncation: Data too long for column' errors when importing data or corruption of non-supported characters.
- The database URL in the example below contains the `sessionVariables=default_storage_engine=InnoDB` parameter. We strongly recommend adding this parameter to avoid data corruption.

```
<jira-database-config>
  <name>defaultDS</name>
  <delegator-name>default</delegator-name>
  <database-type>mysql57</database-type>
  <jdbc-datasource>
    <url>jdbc:mysql://dbserver:3306/jiradb?useUnicode=true&characterEncoding=UTF8&
sessionVariables=default_storage_engine=InnoDB</url>
    <driver-class>com.mysql.jdbc.Driver</driver-class>
    <username>jiradbuser</username>
    <password>password</password>
    <pool-min-size>20</pool-min-size>
    <pool-max-size>20</pool-max-size>
    <pool-max-wait>30000</pool-max-wait>
    <pool-max-idle>20</pool-max-idle>
    <pool-remove-abandoned>true</pool-remove-abandoned>
    <pool-remove-abandoned-timeout>300</pool-remove-abandoned-timeout>

    <validation-query>select 1</validation-query>
    <min-evictable-idle-time-millis>60000</min-evictable-idle-time-millis>
    <time-between-eviction-runs-millis>300000</time-between-eviction-runs-millis>

    <pool-test-while-idle>true</pool-test-while-idle>
    <pool-test-on-borrow>false</pool-test-on-borrow>
    <validation-query-timeout>3</validation-query-timeout>
  </jdbc-datasource>
</jira-database-config>
```

## Known issues

Here's a list of known issues for this database. Expand each of them for more details.

The database health check displays the following warning despite configuring Jira to use MySQL 5.7:

```
Your mysql database is currently using an unsupported collation: utf8mb4_bin. You should change this to a supported collation: utf8_bin
```

### Solution

This problem might occur if the `dbconfig.xml` file doesn't get updated with the proper database type, still using `mysql` instead of `mysql57`. You can fix this problem by manually editing the file:

1. Go to Jira home directory, and edit the `dbconfig.xml` file.
2. Change the `database-type` to `mysql57`.

You can also change this by using the Jira configuration tool, as described [above](#).

If you grant permissions in MySQL to a hostname such as `localhost`, then you'll need to use the same string when connecting to the database from Jira. Using `127.0.0.1` won't work, even though it resolves to the same place. This mistake will result in warnings about tables not being found, because the JDBC connection didn't have permissions to create the new tables when Jira was set up.

If you are using a MySQL database with any of the following, you may experience problems with your connections dropping out (see [JRA-15731](#) for details):

- Jira 3.13 or later,
- version 5.5.25 or higher of Tomcat 5,
- version 6.0.13 or higher of Tomcat 6,

For more info on how to address this, see [Surviving connection closures](#).

Special characters for database password are not supported, because Jira can't interpret them.

Jira uses the READ-COMMITTED transaction isolation level with MySQL, which currently supports only row-based binary logging.

If you require MySQL's binary logging features, you must configure MySQL's binary logging format to be 'row-based'. Otherwise, you may encounter problems when creating issues in Jira.

# Connecting Jira applications to MySQL 8.0

These instructions will help you connect Jira to a supported MySQL database.

- [Before you begin](#)
- [Create and configure the MySQL database](#)
- [Copy the MySQL JDBC driver](#)
- [Configure Jira to connect to the database](#)
- [Start Jira](#)
- [Database connection fields](#)
- [Known issues](#)

## Before you begin

Here is some prerequisite information you should know about:

- Check [known issues](#).
- If you are [migrating Jira to another server](#), create an export of your data as an [XML backup](#). You will then be able to transfer data from your old database to your new database, as described in [Switching databases](#).
- If you plan to set up Confluence and Jira on the same MySQL server, read the [Confluence MySQL setup guide](#). Confluence requirements are more strict than Jira's, so you should configure MySQL to suit Confluence. This configuration will work for Jira, too.
- Shut down Jira before you begin, unless you are running the setup wizard.
- In this guide, we recommend that you use the `utf8mb4_bin` collation, however `utf8mb4_0900_ai_ci`, which is the default collation for MySQL 8.0, is also supported. Note that if you use MySQL on Amazon RDS, you'll get the default collation unless you specifically change it.

## 1. Create and configure the MySQL database

 When creating the database, remember your **database name**, **user name**, and **port number**, because you'll need them later to connect Jira to your database.

1. Create a database user which Jira will connect as, for example **jiradbuser**.

```
CREATE USER '<USERNAME>' IDENTIFIED BY '<PASSWORD>';
```

2. Create a database for Jira to store issues in, for example **jiradb**.

The database must have a character set of UTF8. To set it, enter the following command from within the MySQL command client:

```
CREATE DATABASE jiradb CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
```

3. Make sure the user has permission to connect to the database, and permission to create and populate tables. You can provide these permissions with the following commands.

```
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,REFERENCES,ALTER,INDEX on <JIRADB>.* TO  
'<USERNAME>'@'<JIRA_SERVER_HOSTNAME>';  
flush privileges;
```

4. Edit the `my.cnf` or `my.ini` (Windows) file in your MySQL Server (for detailed instructions on editing these files, see [MySQL Option Files](#)).
5. Locate the `[mysqld]` section in the file, and add or modify the following parameters:

- Set the default storage engine to InnoDB:

```
[mysqld]
...
default-storage-engine=INNODB
...
```

- Specify the character set used by the database server:

```
[mysqld]
...
character_set_server=utf8mb4
...
```

- Set the default row format to `DYNAMIC`:

```
[mysqld]
...
innodb_default_row_format=DYNAMIC
...
```

- Specify the value of `innodb_log_file_size` to be at least 2G:

```
[mysqld]
...
innodb_log_file_size=2G
...
```

- Ensure the `sql_mode` parameter does not specify `NO_AUTO_VALUE_ON_ZERO`

```
// remove this if it exists
sql_mode = NO_AUTO_VALUE_ON_ZERO
```

6. Restart your MySQL server for the changes to take effect.

Use the Windows Services manager to restart the service.

Run one of the following commands, depending on your setup:

```
/etc/init.d/mysqld stop
```

```
/etc/init.d/mysql stop
```

```
service mysqld stop
```

Then, run the same command, replacing `stop` with `start`.

## 2. Copy the MySQL JDBC driver

Copy the MySQL JDBC driver to the Jira installation directory.

1. Download the recommended MySQL driver [JDBC Connector/J 8.0](#).
2. Copy the driver to the following directory:

```
<Jira-installation-directory>/lib
```

**i** If you are installing Jira using the Windows installer, you will need to do this step after running the Windows installer, but **before** [running the setup wizard](#).

3. Restart the Jira service.
4. If you are installing Jira, skip the rest of the instructions on this page and access Jira in your browser to [run the setup wizard](#) instead.

### 3. Configure Jira to connect to the database

There are two ways to configure your Jira server to connect to your MySQL database:

#### Setup wizard

Use the setup wizard if you have just installed Jira, and are setting it up for the first time. Your settings will be saved to the `dbconfig.xml` file in your [Jira home directory](#).

The [Jira setup wizard](#) will display when you access Jira for the first time in your browser.

1. In the first screen, 'Configure Language and Database', set **Database Connection to My own database**.
2. Set **Database Type to MySQL 8.0**.
3. Fill out the fields, as described in the [Database connection fields](#) section below.
4. Test your connection and save.

#### Configuration tool

Use the configuration tool if you have an existing Jira instance. Your settings will be saved to the `dbconfig.xml` file in your [Jira home directory](#).

1. Run the Jira configuration tool as follows:
  - **Windows:** Open a command prompt and run `config.bat` in the `bin` sub-directory of the [Jira installation directory](#).
  - **Linux/Unix:** Open a console and execute `config.sh` in the `bin` sub-directory of the [Jira installation directory](#).

**i** This command might fail with the error as described in [Unable to Start Jira applications Config Tool due to No X11 DISPLAY variable was set error](#). If it happens, refer to this article for the workaround.

2. Navigate to the **Database tab** and set **Database type to MySQL 8.0**.
3. Fill out the fields, as described in the [Database connection fields](#) section below.
4. Test your connection and save.
5. Restart Jira.

### 4. Start Jira

You should now have Jira configured to connect to your MySQL database. The next step is to start it up!

## Database connection fields

The table below explains database connection fields that you can find in the setup wizard, Jira configuration tool, or the `dbconfig.xml` file.

Setup wizard / Configuration tool	Description	dbconfig.xml
<b>Hostname</b>	The name or IP address of the machine that the MySQL server is installed on.	<p>Located in the <code>&lt;url&gt;</code> tag. In the example below, <b>dbserver</b>.</p> <pre>&lt;url&gt;jdbc:mysql://dbserver:3306/jiradb? useUnicode=true&amp; characterEncoding=UTF8&amp; sessionVariables=default_storage_engine=InnoDB &lt;/url&gt;</pre> <p>If you use an IPv6 address, the URL needs to look like this:</p> <pre>&lt;url&gt;jdbc:mysql://address=(protocol=tcp) (host=dbserver)(port=3306)/jiradb? useUnicode=true&amp; characterEncoding=UTF8&amp; sessionVariables=default_storage_engine=InnoDB &lt;/url&gt;</pre>
<b>Port</b>	The TCP/IP port that the MySQL server is listening on. You can leave this blank to use the default port.	<p>Located in the <code>&lt;url&gt;</code> tag. In the example below, <b>3306</b>.</p> <pre>&lt;url&gt;jdbc:mysql://dbserver:3306/jiradb? useUnicode=true&amp; characterEncoding=UTF8&amp; sessionVariables=default_storage_engine=InnoDB &lt;/url&gt;</pre>
<b>Database</b>	The name of your MySQL database (into which Jira will save its data). You should have created this in <a href="#">Step 1</a> above.	<p>Located in the <code>&lt;url&gt;</code> tag. In the example below, <b>jiradb</b>.</p> <pre>&lt;url&gt;jdbc:mysql://dbserver:3306/jiradb ? useUnicode=true&amp; characterEncoding=UTF8&amp; sessionVariables=default_storage_engine=InnoDB &lt;/url&gt;</pre>
<b>Username</b>	The user that Jira uses to connect to the MySQL server. You should have created this in <a href="#">Step 1</a> above.	<pre>&lt;username&gt;jiradbuser&lt;/username&gt;</pre>
<b>Password</b>	The user's password used to authenticate with the MySQL server.	<pre>&lt;password&gt;jiradbuser&lt;/password&gt;</pre>

- For more information about the child elements of `<jdbc-datasource/>` beginning with `pool` in the `dbconfig.xml` file above, see [Tuning database connections](#).

- Both the Jira setup wizard and database configuration tool also add the element `<validation-query>select 1</validation-query>` to this file, which is usually required when running Jira with default MySQL installations. See [Surviving connection closures](#) for more information.
- The database URL in the example below assumes a UTF-8 database i.e. that your database was created using a command similar to `create database jiradb character set utf8`; If you do not specify `character set utf8` when creating this database, you risk getting 'Data truncation: Data too long for column' errors when importing data or corruption of non-supported characters.
- The database URL in the example below contains the `sessionVariables=default_storage_engine=InnoDB` parameter. We strongly recommend adding this parameter to avoid data corruption.

```
<jira-database-config>
<name>defaultDS</name>
<delegator-name>default</delegator-name>
<database-type>mysql8</database-type>
<jdbc-datasource>
  <url>jdbc:mysql://dbserver:3306/jiradb?useUnicode=true&amp;characterEncoding=UTF8&amp;
sessionVariables=default_storage_engine=InnoDB</url>
  <driver-class>com.mysql.cj.jdbc.Driver</driver-class>
  <username>jiradbuser</username>
  <password>password</password>
  <pool-min-size>20</pool-min-size>
  <pool-max-size>20</pool-max-size>
  <pool-max-wait>30000</pool-max-wait>
  <pool-max-idle>20</pool-max-idle>
  <pool-remove-abandoned>true</pool-remove-abandoned>
  <pool-remove-abandoned-timeout>300</pool-remove-abandoned-timeout>

  <validation-query>select 1</validation-query>
  <min-evictable-idle-time-millis>60000</min-evictable-idle-time-millis>
  <time-between-eviction-runs-millis>300000</time-between-eviction-runs-millis>

  <pool-test-while-idle>true</pool-test-while-idle>
  <pool-test-on-borrow>false</pool-test-on-borrow>
  <validation-query-timeout>3</validation-query-timeout>
</jdbc-datasource>
</jira-database-config>
```

## Known issues

Here's a list of known issues for this database. Expand each of them for more details.

The database health check displays the following warning despite configuring Jira to use MySQL 8.0:

```
Your mysql database is currently using an unsupported collation: <collation>. You should change this
to a supported collation: utf8mb4_bin
```

### Solution

This problem might occur if the `dbconfig.xml` file doesn't get updated with the proper database type, still using `mysql` instead of `mysql8`. You can fix this problem by manually editing the file:

1. Go to Jira home directory, and edit the `dbconfig.xml` file.
2. Change the `database-type` to `mysql8`.

You can also change this by using the Jira configuration tool, as described [above](#).

If the problem keeps occurring, check that the collation used in the MySQL server is set to `utf8mb4_bin`.

When running Jira with MySQL 8.0, you might encounter the following problem related to the time zone:

If that happens, you can solve it in two ways:

#### Option 1:

1. Edit the `my.cnf` or `my.ini` (Windows) file in your MySQL server (for more info on editing these files, see [MySQL Option Files](#)).
2. Locate the `[mysqld]` section in the file, and add or modify the following parameters:

```
default_time_zone='+00:00'
```

3. Restart the MySQL server.

#### Option 2:

1. Edit the `dbconfig.xml` file in the Jira home directory.
2. Add `serverTimezone=UTC` to `<url>`, like in the example below:

```
<url>jdbc:mysql://address=(protocol=tcp)(host=0.0.0.0)(port=3306)/jiradb?  
sessionVariables=default_storage_engine=InnoDB&serverTimezone=UTC</url>
```

If you grant permissions in MySQL to a hostname such as `localhost`, then you'll need to use the same string when connecting to the database from Jira. Using `127.0.0.1` won't work, even though it resolves to the same place. This mistake will result in warnings about tables not being found, because the JDBC connection didn't have permissions to create the new tables when Jira was set up.

If you are using a MySQL database with any of the following, you may experience problems with your connections dropping out (see [JRA-15731](#) for details):

- Jira 3.13 or later,
- version 5.5.25 or higher of Tomcat 5,
- version 6.0.13 or higher of Tomcat 6,

For more info on how to address this, see [Surviving connection closures](#).

Special characters for database password are not supported, because Jira can't interpret them.

Jira uses the `READ-COMMITTED` transaction isolation level with MySQL, which currently supports only row-based binary logging.

If you require MySQL's binary logging features, you must configure MySQL's binary logging format to be 'row-based'. Otherwise, you may encounter problems when creating issues in Jira.

# Connecting Jira applications to Oracle

These instructions will help you connect Jira to an [Oracle](#) database.

## On this page:

- [Before you begin](#)
- [1. Configure Oracle](#)
- [2. Download the Oracle JDBC driver](#)
- [3. Configure your Jira Server to connect to your Oracle database](#)
- [4. Start Jira](#)
- [Known issues and troubleshooting](#)

## Before you begin

- Check whether your version of Oracle is supported. See [Supported platforms](#).
- Check which version of the JDBC driver you need to download for your Oracle version. See [Supported platforms](#).
- Check [known issues and troubleshooting](#) below.
- If you are [migrating Jira to another server](#), create an export of your data as a [backup](#). You will then be able to transfer data from your old database to your new database as described in [Switching databases](#).
- Shut down Jira before you begin unless you are running the setup wizard.

## 1. Configure Oracle

1. Ensure that you have a database instance available for Jira either create a new one or use an existing one.
2. Within that database instance, create a user which Jira will connect to. For example, `jiradbuser`. Make sure that you remember this database user name as it will be used to configure Jira's connection to this database in the following steps.

```
create user <user> identified by <user_pass> default tablespace
tablespace_name> quota unlimited on <tablespace_name>;
```

 When you create a user in Oracle:

- Oracle will create a "schema" automatically
- the tablespace for the table objects must be specified

3. Ensure that the user has the following privileges:

```
grant connect to <user>;
grant create table to <user>;
grant create sequence to <user>;
grant create trigger to <user>;
```

 It is critically important that **the user is granted the exact privileges as indicated above**. Jira requires only these privileges if either less or more than these privileges are applied, some Jira functions may not work properly.

Simply put, for Jira functions to work as expected, we advise that you **grant specific privileges** to the user, and **not assign a role** to the user.

For example, if you grant the `RESOURCE` role to a user, and the `RESOURCE` role grants the `SELECT ANY TABLE` privilege, then Jira functions may not work as expected. Thus, we recommend that you grant the exact privileges to the user instead.

4. Ensure that the database is configured to use the same character encoding as Jira. The recommended encoding is `AL32UTF8` (the Oracle equivalent of Unicode UTF-8).

## 2. Download the Oracle JDBC driver

Download the right JDBC driver for your Oracle version:

1. [Download the Oracle JDBC driver.](#)
2. Copy the downloaded .jarfile to the lib/ directory in the Jira installation directory.

For more information on supported drivers, see [Supported platforms](#).

### 3. Configure your JiraServer to connect to your Oracle database

There are two ways to configure your JIRA server to connect to your Oracle database:

- By using the **Jira** setup wizard. Use this method if you have just installed Jira and are setting it up for the first time. Your settings will be saved to the dbconfig.xml file in your [Jira home directory](#).
- By using the **Jira** configuration tool. Use this method if you have an existing JIRA instance. The settings will be saved to the dbconfig.xml file in your [Jira home directory](#).

#### Configuring Jiraby using the setup wizard

You'll see the [Jira setup wizard](#) when you access **Jira** for the first time in your browser.

1. On the **Configure Language and Database** screen that appears first, set **Database Connection** to **My own database**.
2. Set **Database Type** to **Oracle**.
3. Fill out the fields as described in the [Database connection fields](#) section.
4. Test your connection and save.

#### Configuring Jira by using the configuration tool

1. Run the Jira configuration tool as follows:
  - **Windows:** Open a command prompt and run config.bat in the bin sub-directory of the [Jira installation directory](#).
  - **Linux/Unix:** Open a console and execute config.sh in the bin sub-directory of the [Jira installation directory](#).

 This command might fail with the error as described in [Unable to Start Jira applications Config Tool due to No X11 DISPLAY variable was set error](#). If it happens, refer to this article for the workaround.

2. Go to the **Database** tab and set **Database type** to **Oracle**.
3. Fill out the fields as described in the [Database connection fields](#) section.
4. Test your connection and save. Any custom settings specified while manually configuring Jira with Oracle (e.g., adding the <connection-properties>SetBigStringTryClob=true</connection-properties>) will be deleted. You will need to reinstate them manually.
5. Restart Jira.

#### Database connection fields

Setup Wizard / Configuration tool	dbconfig.xml	Description
<b>Hostname</b>	Located in the <url> tag (bold text in the following example): <code>&lt;url&gt;jdbc:oracle:thin:@<b>dbserver</b>:1521/ORCL&lt;/url&gt;</code>	The name or IP address of the machine that the Oracle server is installed on.

<b>Port</b>	Located in the <code>&lt;url&gt;</code> tag (bold text in example below): <code>&lt;url&gt;jdbc:oracle:thin:@dbserver: <b>1521</b>/ORCL&lt;/url&gt;</code>	The TCP/IP port that the Oracle server is listening on. The default port number for Oracle is "1521".
<b>SID</b>	Located in the <code>&lt;url&gt;</code> tag (bold text in example below): <code>&lt;url&gt;jdbc:oracle:thin:@dbserver:1521/<b>ORCL</b> &lt;/url&gt;</code>	The Oracle "System Identifier". The default value for most Oracle servers is "ORCL". If you are using the Oracle Express Edition, the default value will be "XE".
<b>Username</b>	Located in the <code>&lt;username&gt;</code> tag (see bold text in example below): <code>&lt;username&gt; <b>jiradbuser</b> &lt;/username&gt;</code>	The user that Jira uses to connect to the Oracle server. You should have created this user in <a href="#">Step 1</a> of this guide.
<b>Password</b>	Located in the <code>&lt;password&gt;</code> tag (see bold text in example below): <code>&lt;password&gt; <b>jiradbuser</b> &lt;/password&gt;</code>	The user's password that's used for authentication with the Oracle server.

### Sample dbconfig.xml file

For more information about the child elements of `<jdbc-datasource/>` beginning with `pool` in the `dbconfig.xml` file above, see [Tuning database connections](#).

```
<jira-database-config>
  <name>defaultDS</name>
  <delegator-name>default</delegator-name>
  <database-type>oracle10g</database-type>
  <jdbc-datasource>
    <url>jdbc:oracle:thin:@dbserver:1521/ORCL</url>
    <driver-class>oracle.jdbc.OracleDriver</driver-class>
    <username>jiradbuser</username>
    <password>password</password>
    <pool-min-size>20</pool-min-size>
    <pool-max-size>20</pool-max-size>
    <pool-max-wait>30000</pool-max-wait>
    <pool-max-idle>20</pool-max-idle>
    <pool-remove-abandoned>true</pool-remove-abandoned>
    <pool-remove-abandoned-timeout>300</pool-remove-abandoned-timeout>

    <validation-query>select 1 from dual</validation-query>
    <min-evictable-idle-time-millis>60000</min-evictable-idle-time-millis>
    <time-between-eviction-runs-millis>300000</time-between-eviction-runs-millis>

    <pool-test-while-idle>true</pool-test-while-idle>
    <pool-test-on-borrow>false</pool-test-on-borrow>
  </jdbc-datasource>
</jira-database-config>
```

## 4. Start Jira

You should now have Jira configured to connect to your Oracle database. The next step is to start it up!

Congratulations, you now have Jira connected to your Oracle database.

### Known issues and troubleshooting

If you face problems when dealing with custom workflows or working with issues that have long descriptions, comments or custom field values, try adding the element `<connection-properties>SetBigStringTryClob=true</connection-properties>` as a child of the `</jdbc-datasource>` element in your `dbconfig.xml` file.

This connection property may solve the encountered problems. You'll need to restart Jira afterward.

# Connecting Jira applications to SQL Server 2016

These instructions will help you connect Jira to a Microsoft SQL Server 2016 database.

Before you begin

- If you're [Migrating Jira applications to another server](#), create an export of your data as an [backup](#). You will then be able to transfer data from your old database to your new database, as described in [Switching databases](#).
- Stop Jira before you begin, unless you just started the installation and are running the Setup Wizard.

## 1. Create and configure the SQL Server database

**i** When creating the database, remember your **database name**, **user name**, **schema name**, and **port number**, because you'll need them later to connect Jira to your database.

### 1. Create a database for Jira (e.g. `jiradb`).

- Make sure the collation type is **case-insensitive**.

We support `SQL_Latin1_General_CP437_CI_AI` and `Latin1_General_CI_AI` as case-insensitive, accent-insensitive, and language neutral collation types. If your SQL Server installation's collation type settings have not been changed from their defaults, check the collation type settings.

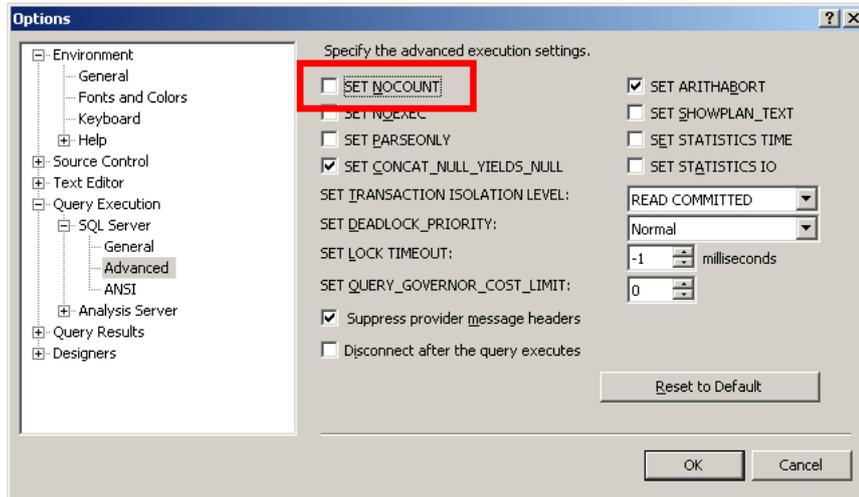
- SQL Server uses Unicode encoding to store characters. This is sufficient to prevent any possible encoding problems.
2. Create a database user which Jira will connect as (e.g. `jiradbuser`). This user should *not* be the database owner, but *should* be in the `db_owner` role.
  3. Create an empty 'schema' in the database for the Jira tables (e.g. `jiraschema`).

A 'schema' in SQL Server 2016 is a distinct namespace used to contain objects and is *different* from a traditional database schema. You are not required to create any of Jira's tables, fields or relationships (Jira will create these objects in your empty schema when it starts for the first time). You can read more on SQL Server 2016 schemas in the relevant [Microsoft documentation](#).

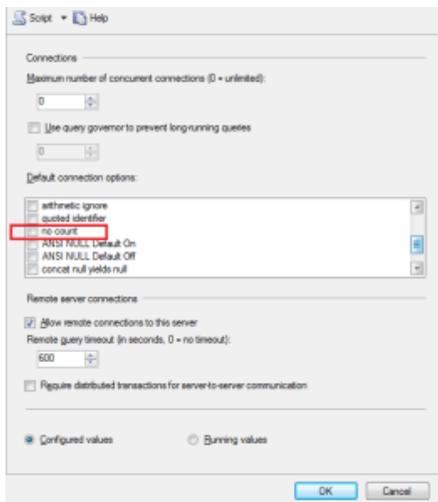
4. Make sure that the database user has permission to connect to the database, and to create and populate tables in the newly-created schema.
5. Make sure that TCP/IP is enabled on SQL Server and is listening on the correct port. A default SQL Server installation uses port number 1433.
6. Make sure that SQL Server is operating in the appropriate authentication mode.

By default, SQL Server operates in 'Windows Authentication Mode'. However, if your user is not associated with a trusted SQL connection, i.e. 'Microsoft SQL Server, Error: 18452' is received during Jira startup, you will need to change the authentication mode to 'Mixed Authentication Mode'. Read the Microsoft documentation on authentication modes and changing the authentication mode to 'Mixed Authentication Mode'

7. Turn off the NOCOUNT option.
  - a. Open SQL Server Management Studio.
  - b. Go to **Tools > Options > Query Execution > SQL Server > Advanced**, and clear the **NOCOUNT** check box.



- c. Right-click your server in the Object Explorer, and go to **Properties > Connections > Default Connections**. Clear the **no count** option.



8. Access the **Query Console** by right clicking on the newly created database and selecting 'New Query'. Run the following command to set the isolation level.

```
ALTER DATABASE THE-NEW-DATABASE-CREATED-FOR-JIRA READ_COMMITTED_SNAPSHOT ON
```

## 2. Configure Jira to connect to the database

There are two ways to configure your Jira server to connect to your SQL Server database.

- **Using the Jira setup wizard** Use this method, if you have just installed Jira and are setting it up for the first time. Your settings will be saved to the `dbconfig.xml` file in your [Jira application home directory](#).

The [Jira setup wizard](#) will display when you access Jira for the first time in your browser.

1. In the first screen, 'Configure Language and Database', set **Database Connection to My own database**.
2. Set **Database Type to SQL Server**.
3. Fill out the fields, as described in the [Database connection fields](#) section below.
4. Test your connection and save.

- **Using the Jira configuration tool** Use this method, if you have an existing Jira instance. Your settings will be saved to the `dbconfig.xml` file in your [Jira application home directory](#).

1. Run the Jira configuration tool as follows:
  - **Windows:** Open a command prompt and run `config.bat` in the `bin` sub-directory of the [Jira installation directory](#).

- **Linux/Unix:** Open a console and execute `config.sh` in the `bin` sub-directory of the [Jira installation directory](#).

 This command might fail with the error as described in [Unable to Start Jira applications Config Tool due to No X11 DISPLAY variable was set error](#). If it happens, refer to this article for the workaround.

2. Navigate to the **Database** tab and set **Database type** to **SQL Server**.
3. Fill out the fields, as described in the [Database connection fields](#) section below.
4. Test your connection and save.
5. Restart Jira.

### Database connection fields

The table shows the fields you'll need to fill out when connecting Jira to your database. You can also refer to them, and the sample `dbconfig.xml` file below, if you'd like to create or edit the `dbconfig.xml` file manually.

Setup Wizard / Configuration Tool	dbconfig.xml	Description
<b>Hostname</b>	Located in the <code>&lt;url&gt;</code> tag (bold text in example below): <code>&lt;url&gt;jdbc:sqlserver://<b>dbserver</b>:1433;databaseName=jiradb&lt;/url&gt;</code>	The name or IP address of the machine that the SQL Server server is installed on.
<b>Port</b>	Located in the <code>&lt;url&gt;</code> tag (bold text in example below): <code>&lt;url&gt;jdbc:sqlserver://dbserver :<b>1433</b>;databaseName=jiradb&lt;/url&gt;</code>	The TCP/IP port that the SQL Server server is listening on. You can leave this blank to use the default port.
<b>Database</b>	Located in the <code>&lt;url&gt;</code> tag (bold text in example below): <code>&lt;url&gt;jdbc:sqlserver://dbserver :1433;databaseName=<b>jiradb</b>&lt;/url&gt;</code>	The name of your SQL Server database (into which Jira will save its data). You should have created this in <a href="#">Step 1</a> above.
<b>Username</b>	Located in the <code>&lt;username&gt;</code> tag (see bold text in example below): <code>&lt;username&gt; <b>jiradbuser</b> &lt;/username&gt;</code>	The user that Jira uses to connect to the SQL Server server. You should have created this in <a href="#">Step 1</a> above.
<b>Password</b>	Located in the <code>&lt;password&gt;</code> tag (see bold text in example below): <code>&lt;password&gt; <b>jiradbuser</b> &lt;/password&gt;</code>	The user's password used to authenticate with the SQL Server server.
<b>Schema</b>	Located in the <code>&lt;schema-name&gt;</code> tag (see bold text in example below): <code>&lt;schema-name&gt; <b>dbo</b> &lt;/schema-name&gt;</code>	The name of the schema that your SQL Server database uses. You should have created this in <a href="#">Step 1</a> above.

### Sample dbconfig.xml file

For more information about the child elements of `<jdbc-datasource/>` beginning with `pool` in the `dbconfig.xml` file above, see [Tuning database connections](#).

```

<jira-database-config>
<name>defaultDS</name>
<delegator-name>default</delegator-name>
<database-type>mssql</database-type>
<schema-name>jiraschema</schema-name>
<jdbc-datasource>
  <url>jdbc:sqlserver://dbserver:1433;databaseName=jiradb</url>
  <driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>
  <username>jiradbuser</username>
  <password>password</password>
  <pool-min-size>20</pool-min-size>
  <pool-max-size>20</pool-max-size>
  <pool-max-wait>30000</pool-max-wait>
  <pool-max-idle>20</pool-max-idle>
  <pool-remove-abandoned>true</pool-remove-abandoned>
  <pool-remove-abandoned-timeout>300</pool-remove-abandoned-timeout>

  <validation-query>select 1</validation-query>
  <min-evictable-idle-time-millis>60000</min-evictable-idle-time-millis>
  <time-between-eviction-runs-millis>300000</time-between-eviction-runs-millis>

  <pool-test-while-idle>true</pool-test-while-idle>
  <pool-test-on-borrow>false</pool-test-on-borrow>
</jdbc-datasource>
</jira-database-config>

```

### 3. Schedule regular database maintenance tasks

To achieve and maintain optimal MS SQL performance, schedule daily maintenance tasks to update database statistics.

#### Schedule a daily maintenance task for hot tables

Hot tables are the most active tables in your database. For example, `propertyentry`, `changeitem`, and `changegroup` are large data tables that are used frequently and require regular updating of statistics.

To set up a daily maintenance task for hot tables, run the following command:

```
UPDATE STATISTICS <table.name>
```

#### Schedule a weekly maintenance task for the whole database

To set up a weekly maintenance task for the whole database, run the following command:

```
UPDATE STATISTICS with fullscan
```



For large databases, updating statistics with `fullscan` might take a long time to complete. To minimize the impact on your production environment, schedule this maintenance task for off-peak hours.

For more information on how to update MS SQL Server statistics, see the [official Microsoft documentation](#).

## 4. Start Jira

You should now have Jira configured to connect to your SQL Server database. The next step is to start it up!

# Connecting Jira applications to SQL Server 2017

These instructions will help you connect Jira to a Microsoft SQL Server 2017 database.

Before you begin

- If you're [Migrating Jira applications to another server](#), create an export of your data as an [backup](#). You will then be able to transfer data from your old database to your new database, as described in [Switching databases](#).
- Stop Jira before you begin, unless you just started the installation and are running the Setup Wizard.

## 1. Create and configure the SQL Server database

**i** When creating the database, remember your **database name**, **user name**, **schema name**, and **port number**, because you'll need them later to connect Jira to your database.

### 1. Create a database for Jira (e.g. `jiradb`).

- Make sure the collation type is **case-insensitive**.

We support `SQL_Latin1_General_CP437_CI_AI` and `Latin1_General_CI_AI` as case-insensitive, accent-insensitive, and language neutral collation types. If your SQL Server installation's collation type settings have not been changed from their defaults, check the collation type settings.

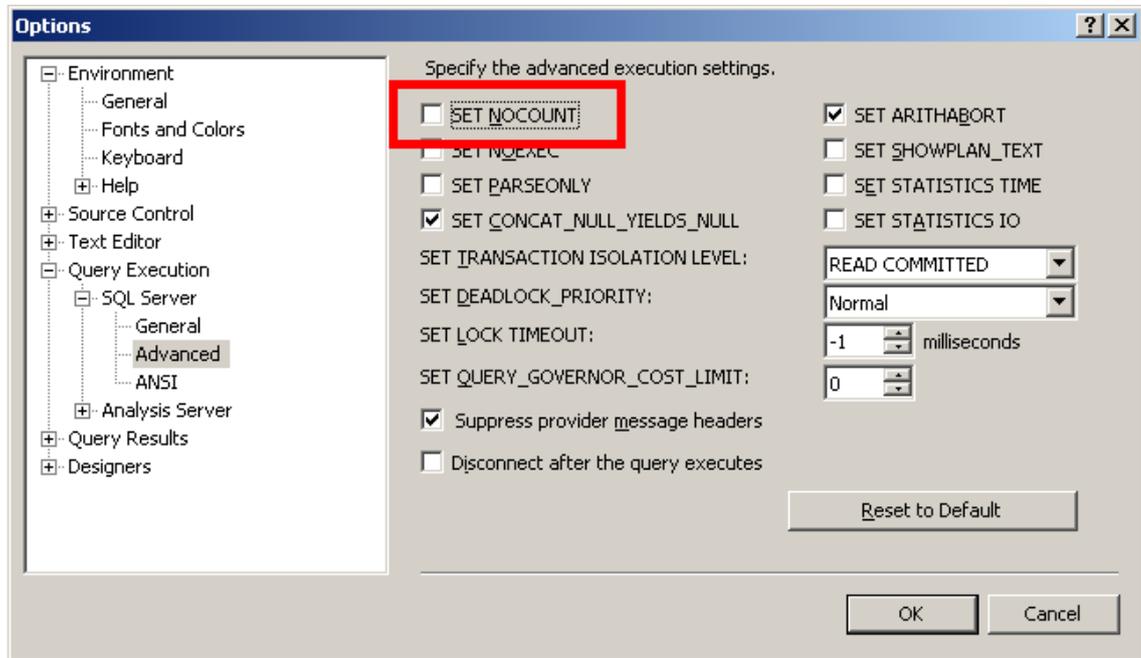
- SQL Server uses Unicode encoding to store characters. This is sufficient to prevent any possible encoding problems.
2. Create a database user which Jira will connect as (e.g. `jiradbuser`). This user should *not* be the database owner, but *should* be in the `db_owner` role.
  3. Create an empty 'schema' in the database for the Jira tables (e.g. `jiraschema`).

A 'schema' in SQL Server 2017 is a distinct namespace used to contain objects and is *different* from a traditional database schema. You are not required to create any of Jira's tables, fields or relationships (Jira will create these objects in your empty schema when it starts for the first time). You can read more on SQL Server 2017 schemas in the relevant [Microsoft documentation](#).

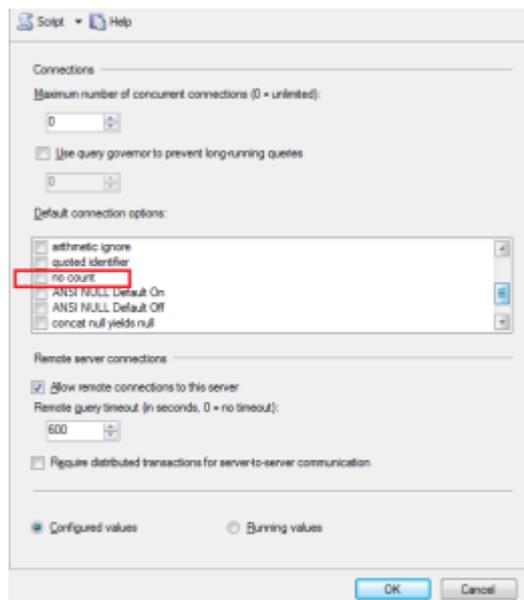
4. Make sure that the database user has permission to connect to the database, and to create and populate tables in the newly-created schema.
5. Make sure that TCP/IP is enabled on SQL Server and is listening on the correct port. A default SQL Server installation uses port number 1433.
6. Make sure that SQL Server is operating in the appropriate authentication mode.

By default, SQL Server operates in 'Windows Authentication Mode'. However, if your user is not associated with a trusted SQL connection, i.e. 'Microsoft SQL Server, Error: 18452' is received during Jira startup, you will need to change the authentication mode to 'Mixed Authentication Mode'. Read the Microsoft documentation on authentication modes and changing the authentication mode to 'Mixed Authentication Mode'

7. Turn off the NOCOUNT option.
  - a. Open SQL Server Management Studio.
  - b. Go to **Tools > Options > Query Execution > SQL Server > Advanced**, and clear the **NOCOUNT** check box.



- c. Right-click your server in the Object Explorer, and go to **Properties > Connections > Default Connections**. Clear the **no count** option.



8. Access the **Query Console** by right clicking on the newly created database and selecting 'New Query'. Run the following command to set the isolation level.

```
ALTER DATABASE THE-NEW-DATABASE-CREATED-FOR-JIRA READ_COMMITTED_SNAPSHOT ON
```

## 2. Configure Jira to connect to the database

There are two ways to configure your Jira server to connect to your SQL Server database.

- **Using the Jira setup wizard** Use this method, if you have just installed Jira and are setting it up for the first time. Your settings will be saved to the `dbconfig.xml` file in your [Jira application home directory](#).

The [Jira setup wizard](#) will display when you access Jira for the first time in your browser.

1. In the first screen, 'Configure Language and Database', set **Database Connection to My own database**.
2. Set **Database Type to SQL Server**.

3. Fill out the fields, as described in the [Database connection fields](#) section below.
4. Test your connection and save.

- **Using the Jira configuration tool** Use this method, if you have an existing Jira instance. Your settings will be saved to the `dbconfig.xml` file in your [Jira application home directory](#).

1. Run the Jira configuration tool as follows:
  - **Windows:** Open a command prompt and run `config.bat` in the `bin` sub-directory of the [Jira installation directory](#).
  - **Linux/Unix:** Open a console and execute `config.sh` in the `bin` sub-directory of the [Jira installation directory](#).

 This command might fail with the error as described in [Unable to Start Jira applications Config Tool due to No X11 DISPLAY variable was set error](#). If it happens, refer to this article for the workaround.

2. Navigate to the **Database** tab and set **Database type** to **SQL Server**.
3. Fill out the fields, as described in the [Database connection fields](#) section below.
4. Test your connection and save.
5. Restart Jira.

### Database connection fields

The table shows the fields you'll need to fill out when connecting Jira to your database. You can also refer to them, and the sample `dbconfig.xml` file below, if you'd like to create or edit the `dbconfig.xml` file manually.

Setup Wizard / Configuration Tool	dbconfig.xml	Description
<b>Hostname</b>	Located in the <code>&lt;url&gt;</code> tag (bold text in example below): <code>&lt;url&gt;jdbc:sqlserver://<b>dbserver</b> :1433;databaseName=jiradb&lt;/url&gt;</code>	The name or IP address of the machine that the SQL Server server is installed on.
<b>Port</b>	Located in the <code>&lt;url&gt;</code> tag (bold text in example below): <code>&lt;url&gt;jdbc:sqlserver://dbserver :<b>1433</b>;databaseName=jiradb&lt;/url&gt;</code>	The TCP/IP port that the SQL Server server is listening on. You can leave this blank to use the default port.
<b>Database</b>	Located in the <code>&lt;url&gt;</code> tag (bold text in example below): <code>&lt;url&gt;jdbc:sqlserver://dbserver :1433;databaseName=<b>jiradb</b>&lt;/url&gt;</code>	The name of your SQL Server database (into which Jira will save its data). You should have created this in <a href="#">Step 1</a> above.
<b>Username</b>	Located in the <code>&lt;username&gt;</code> tag (see bold text in example below): <code>&lt;username&gt; <b>jiradbuser</b> &lt;/username&gt;</code>	The user that Jira uses to connect to the SQL Server server. You should have created this in <a href="#">Step 1</a> above.
<b>Password</b>	Located in the <code>&lt;password&gt;</code> tag (see bold text in example below): <code>&lt;password&gt; <b>jiradbuser</b> &lt;/password&gt;</code>	The user's password used to authenticate with the SQL Server server.
<b>Schema</b>	Located in the <code>&lt;schema-name&gt;</code> tag (see bold text in example below): <code>&lt;schema-name&gt; <b>dbo</b> &lt;/schema-name&gt;</code>	The name of the schema that your SQL Server database uses. You should have created this in <a href="#">Step 1</a> above.

### Sample dbconfig.xml file

For more information about the child elements of `<jdbc-datasource/>` beginning with `pool` in the `dbconfig.xml` file above, see [Tuning database connections](#).

```
<jira-database-config>
<name>defaultDS</name>
<delegator-name>default</delegator-name>
<database-type>mssql</database-type>
<schema-name>jiraschema</schema-name>
<jdbc-datasource>
  <url>jdbc:sqlserver://dbserver:1433;databaseName=jiradb</url>
  <driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>
  <username>jiradbuser</username>
  <password>password</password>
  <pool-min-size>20</pool-min-size>
  <pool-max-size>20</pool-max-size>
  <pool-max-wait>30000</pool-max-wait>
  <pool-max-idle>20</pool-max-idle>
  <pool-remove-abandoned>true</pool-remove-abandoned>
  <pool-remove-abandoned-timeout>300</pool-remove-abandoned-timeout>

  <validation-query>select 1</validation-query>
  <min-evictable-idle-time-millis>60000</min-evictable-idle-time-millis>
  <time-between-eviction-runs-millis>300000</time-between-eviction-runs-millis>

  <pool-test-while-idle>true</pool-test-while-idle>
  <pool-test-on-borrow>false</pool-test-on-borrow>
</jdbc-datasource>
</jira-database-config>
```

### 3. Schedule regular database maintenance tasks

To achieve and maintain optimal MS SQL performance, schedule daily maintenance tasks to update database statistics.

#### Schedule a daily maintenance task for hot tables

Hot tables are the most active tables in your database. For example, `propertyentry`, `changeitem`, and `changeitem` are large data tables that are used frequently and require regular updating of statistics.

To set up a daily maintenance task for hot tables, run the following command:

```
UPDATE STATISTICS <table.name>
```

#### Schedule a weekly maintenance task for the whole database

To set up a weekly maintenance task for the whole database, run the following command:

```
UPDATE STATISTICS with fullscan
```

**i** For large databases, updating statistics with `fullscan` might take a long time to complete. To minimize the impact on your production environment, schedule this maintenance task for off-peak hours.

For more information on how to update MS SQL Server statistics, see the [official Microsoft documentation](#).

## 4. Start Jira

You should now have Jira configured to connect to your SQL Server database. The next step is to start it up!

# Tuning database connections

Jira uses the database connection pool (DBCP) based on Apache Commons DBCP to manage Jira's access to its underlying database.

In earlier Jiraversions, the database connection pool was handled purely through the Apache Tomcat application server running Jira.

From Jiraversion 4.4, Jira's`sdbconfig.xml` file provides a set of database connection pool settings to Tomcat, which in turn are used by Tomcat to manageJira'sdatabase connection pool.

From JIRA version 5.1, the number of database connection pool settings defined in Jira's`sdbconfig.xml` file has substantially increased.

The information on this page can help you tweak Jira'sdatabase connection pool settings. You can do this by using the [Jira configuration tool](#) or by directly editingJira's`sdbconfig.xml` file, as described [in the following sections](#).

The **Advanced** tab of the Jira Configuration Tool makes it easier to both configure and control Jira's database connection pool. The [Database monitoring](#) page (accessible to Jira system administrators) provides a visual tool for monitoring Jira 'sdatabase connection usage.

## On this page:

- [Connection pool architecture](#)
- [Tuning Jira's database connections](#)
  - [Use Jira configuration tool to start Jira 's database connections](#)
  - [Editing the dbconfig.xml file to start Jira 's database connections](#)
  - [DBCP settings](#)
  - [Monitoring the connection pool](#)

## Connection pool architecture

Whenever Jira needs to access (read from or write to) its database, a database connection is required.

A database connection is a large and complex object that handles all communication between Jiraand its database. As such, database connections are time-consuming to establish and consume a significant amount of memory on both the client (the Jira application) and the database server.

To avoid the impact of creating a new database connection for each Jira's database access request, a pool of pre-established database connections is maintained. Each new database access request made by Jira uses a connection from this pool of pre-established connections, as required. This results in the following:

1. When Jirastarts up, a minimum number of database connections are established in the pool between Jiraand its database.
2. When Jira needs to access its database, it:
  - a. requests a database connection from the pool
  - b. uses this database connection to read from and/or write to its database
  - c. returns the database connection to the pool when finished

If the frequency of Jira's database access requests begins to exceed the number of available database connections in the pool, extra connections are automatically created to handle the load.

Conversely, if the frequency of Jira's database access requests begins to drop below the number of available database connections in the pool, connections can be automatically closed to release resources back to the system.

Modern databases can handle hundreds of connections relatively easily and with sufficient memory. On the client side, however, these connections can consume a significant amount of memory. Hence, it is generally best to limit the number of connections to a much smaller number while having a sufficient number for the application so that it doesn't wait for a connection when it needs one.

## Tuning Jira'sdatabase connections

1. Shut down your Jirainstallation.
2. Proceed with one of the following options:
  - Use the [Jira configuration tool](#) to tune Jira 's database connections.
  - Edit the `dbconfig.xml` file at the root of your [Jira home directory](#) .

### Use Jira configuration tool to start Jira 's database connections

1. Start the Jira configuration tool:
  - **Windows:** Open a command prompt and run `config.bat` in the `bin` sub-directory of the [Jira installation directory](#).
  - **Linux/Unix:** Open a console and execute `config.sh` in the `bin` sub-directory of the [Jira installation directory](#).

**i** This command might fail with the error as described in [Unable to Start Jira applications Config Tool due to No X11 DISPLAY variable was set error](#). If it happens, refer to this article for the workaround.

You may need to set the `JAVA_HOME` environment variable to run the Jira configuration tool. See [Installing Java](#) for details.

2. Once the Jira configuration tool is running, select the **Advanced** tab.

The screenshot shows the 'JIRA Configuration Tool' window with the 'Advanced' tab selected. The 'Scalability and Performance' section has the following settings:

- Maximum Size: 20
- Maximum Idle: [empty]
- Minimum Idle/Size: [empty]
- Initial Size: [empty]
- Maximum Wait Time: [empty]
- Pool Statements:
- Maximum Open Statements: [empty]

The 'Eviction Policy' section has the following settings:

- Validation Query: [empty]
- Validation Query Timeout: [empty]
- Test On Borrow:
- Test On Return:
- Test While Idle:
- Time Between Eviction Runs: [empty]
- Minimum Evictable Idle Time: [empty]
- Remove Abandoned:
- Remove Abandoned Timeout: [empty]

Buttons for 'Save' and 'Close' are visible at the bottom of the window.

3. Refer to the [Connection pool settings](#) for more information about the options on this tab. To specify a value for one of these options, ensure that its leftmost checkbox has been selected first.

**i** Some options on the preceding screenshot are simple checkboxes. Selecting these checkboxes sets the values of their associated options to "true". Conversely, clearing these checkboxes sets the values of their associated options to "false".

4. Save your changes. They will be stored as elements in your `dbconfig.xml` file.

### Editing the `dbconfig.xml` file to start Jira 's database connections

Edit the `dbconfig.xml` file at the root of your [Jira home directory](#):

1. Refer to the [Connection pool settings](#) for more information about the elements you can add to your `dbconfig.xml` file to fine-tune Jira 's database connection.
2. Save your edited `dbconfig.xml` file.
3. Restart your Jirainstallation.

### DBCP settings

- The default values of the settings will be written to the `dbconfig.xml` file after one of the following takes place:
  - Youve run the [Jira setup wizard](#).
  - Youve used the **Advanced** tab of the Jira configuration tool to configure your database connection, even if you havent selected any leftmost checkbox for any option.
- If you see the note when not specified in `dbconfig.xml` for the default value of the setting, it means one of the following:
  - The related element wasnt written to the `dbconfig.xml` file after youve run the [Jira setup wizard](#).
  - The related element was written to the `dbconfig.xml` file in one of the following ways:
    - Manually by the accountable user
    - According to the options on the **Advanced** tab specified by selecting their leftmost checkboxes and setting values for these options.
- If you see the note when not specified in `dbconfig.xml` for the default value of the setting, the system will consider this value even though it may be not present in your `dbconfig.xml` file.

The following table features all connection pool settings and their configuration.

Jira configuration tool "Advanced" tab option	Element identifier	Explanation	Recommendations / Notes	Default value*
Maximum Size	pool-maxsize	The maximum number of database connections that can be open at any time.	This value should be sufficiently large enough that Jira rarely needs to wait for a database connection to become available when Jira requires one.  See the <a href="#">Monitoring the connection pool</a> section for suggestions on how to set this parameter.	20

<b>Maximum Idle</b>	pool-maxidle	The maximum number of database connections that are allowed to remain idle in the pool.	<p>Specifying a negative number sets no limit on the number of database connections that can remain idle.</p> <p>If the value of <b>Minimum Idle/Size</b> (see the following setting) is the same as that of <b>Maximum Size</b> (the previous setting), which is the case by default, then this setting has no effect.</p>	Value of <b>Maximum Size</b>
<b>Minimum Idle/Size</b>	pool-minsize  (minidle)	The minimum number of idle database connections that are kept open at any time.	<p>Having this value set to that of <b>Maximum Size</b> (the previous setting), which is the case by default, means the pool will always have a fixed number of connections and idle connections will never be closed.</p> <p>On very large Jira installations, there may be some benefit in specifying a lower value for this setting than that of <b>Maximum Size</b> to conserve resources.</p>	Value of <b>Maximum Size</b>
<b>Initial Size</b>	pool-init-size	The initial number of database connections opened in the pool.	This setting is not usually configured (other than the default value of 0), since a number of database connections are quickly created when Jira starts up.	0 (when not specified in <code>jdbc-config.xml</code> )
<b>Maximum Wait Time</b>	pool-maxwait	The length of time (in milliseconds) that Jira is allowed to wait for a database connection to become available (while there are no free ones available in the pool), before returning an error.	<p>Specifying a value of "-1" means that Tomcat will wait indefinitely.</p> <p>You should specify a time here which is long enough to allow for any contention spikes, but short enough that users will receive a meaningful error rather than just getting no response or a browser timeout.</p>	30000
<p>Advanced settings</p> <p>Generally, changing the following settings isn't required. Refer to the <a href="#">Apache DBCP documentation</a> if necessary.</p>				
<b>Pool Statements</b>	pool-prepared-statements	Enable the pooling of prepared statements for the database connection pool.	<p>Do <b>not</b> amend the default value of false, as it will cause exceptions. For more information, see</p> <div data-bbox="655 1736 1093 1886" style="border: 1px solid gray; padding: 5px;">  <b>JRASERVER-44908</b> - DBPC configuration pool-prepared-statements leads to Statement Leak <span style="border: 1px solid gray; padding: 2px;">CLOSED</span> </div>	false (when not specified in <code>jdbc-config.xml</code> )

<b>Maximum Open Statements</b>	maximum-open- prepared- statements	The maximum number of open statements that can be allocated from the statement pool at the same time.	Do <b>not</b> amend the default value as it will cause exceptions.	0 (when not specified in <code>indbconfig.xml</code> )
<b>Validation Query</b>	validation- query	The SQL query that will be used to validate connections from this pool. If specified, this query MUST be an SQL SELECT statement that returns at least one row.	See <a href="#">Surviving connection closures</a> for more information.	select 1 (for MySQL)  (otherwise, not specified in <code>indbconfig.xml</code> )
<b>Validation Query Timeout</b>	validation- query- timeout	The length of time (in seconds) that the system should wait for a validation query to succeed before it considers the database connection broken.	<p>The length of time should be quite short as the validation query should be designed to do a minimum amount of work.</p> <p>If you specify a <b>Validation Query</b> above, then you should specify a value for the <b>Validation Query Timeout</b> too. If not, a value of "-1" is assumed, which results in the system waiting indefinitely until a validation query succeeds against a broken database connection, which it never will.</p> <div data-bbox="655 1350 1093 1608" style="border: 1px solid red; padding: 5px; margin-top: 10px;"> <p> This should only be done for MySQL. Using a <b>Validation Query Timeout</b> on any database other than MySQL will cause significant problems with the Jira instance.</p> </div>	3 (for MySQL)  (otherwise, not specified in <code>indbconfig.xml</code> )

<p><b>Test On Borrow</b></p>	<p>pool - test - on - borrow</p>	<p>Tests if the database connection is valid when it is borrowed from the database connection pool by Jira.</p> <p>If the database connection is broken, it is removed from the pool.</p>	<p>This value should always be "false" as Jira borrows a connection for each database operation.</p> <div style="border: 1px solid green; padding: 5px; margin-top: 10px;"> <p> If you continue to have problems with database connections closing, try setting this option to "true". However, this should only be used as a last resort and only in the event that decreasing the value of <b>Time Between Eviction Runs</b> has not reduced or prevented problems with database connections closing.</p> </div>	<p>True (when not specified in <code>dbconfig.xml</code>), however this does not take effect unless a Validation Query has been explicitly specified, except for MySQL, which has a default Validation Query, and it will therefore have an effect.</p>
<p><b>Test On Return</b></p>	<p>pool - test - on - return</p>	<p>Tests if the database connection is valid when it is returned to the database connection pool by Jira.</p> <p>If the database connection is broken, it is removed from the pool.</p>	<p>This value should always be "false" as Jira returns borrowed connections for each database operation.</p>	<p>false (when not specified in <code>dbconfig.xml</code>)</p>
<p><b>Test While Idle</b></p>	<p>pool - test - while - idle</p>	<p>Periodically tests if the database connection is valid when it is idle.</p> <p>If the database connection is broken, it is removed from the pool.</p>	<p>This should be set to "true" for MySQL.</p> <p>By default, MySQL database servers close database connections if they are not used for an extended period of time. This causes problems with Jira installations (which use MySQL databases) that are largely inactive for long periods, for example overnight. Setting this to "true" will work around this behavior.</p> <p><b>Test While Idle</b> only needs to be specified if you have specified a <b>Validation Query</b> above.</p>	<p>true (for MySQL)</p> <p>false (when not specified in <code>dbconfig.xml</code>)</p>

<b>Time Between Eviction Runs</b>	time - between eviction runs - milliseconds	<p>The number of milliseconds to sleep between runs of the idle object eviction thread. When non-positive, no idle object eviction thread will be run.</p> <p>The eviction thread will remove idle database connections when the number of idle connections exceeds <b>Minimum Idle/Size</b> (above).</p>	<p>This should be set to a positive but largish value for MySQL so the evictor runs and tests connections. A reasonable value would be 300000 (5 minutes).</p> <div data-bbox="655 315 1091 510" style="border: 1px solid green; padding: 5px; margin-top: 10px;"> <p> If you continue to have problems with database connections closing, try setting this option to a lower value.</p> </div>	<p>300000 (for MySQL)</p> <p>5000 (for HSQLDB)</p> <p>(otherwise, not specified in <code>indbconfig.xml</code>)</p>
<b>Minimum Evictable Idle Time</b>	minimum - evictable - idle - time - milliseconds	<p>The minimum amount of time an object may sit idle in the database connection pool before it is eligible for eviction by the idle object eviction (if any).</p>		<p>60000 (for MySQL)</p> <p>4000 (for HSQLDB)</p> <p>(otherwise, not specified in <code>indbconfig.xml</code>)</p>
<b>Remove Abandoned</b>	pool - remove - abandoned	<p>Flag to remove abandoned database connections if they exceed the <b>Removed Abandoned Timeout</b> (below).</p> <p>If an internal failure occurs, it is possible that Jira may borrow a connection and never return it. If this happens too often, then the pool may run short of database connections, causing Jira's performance to degrade or Jira to fail altogether.</p>	<p>This value should be set to "true".</p> <p>This will allow the pool to recover any abandoned connections and prevent this from affecting system performance.</p>	<p>true</p>

<b>Remove Abandoned Timeout</b>	pool - remainder - abandoned-timeout	The length of time (in seconds) that a database connection can be idle before it is considered abandoned.		300
---------------------------------	--------------------------------------	---	--	-----

### Monitoring the connection pool

Jira provides a view of its database connection usage via the **Database Monitoring** page. See [Monitoring database connection usage](#) for more information.

# Surviving connection closures

When a database server reboots or a network failure has occurred, all connections in the database connection pool are broken. To overcome this issue, Jirawould normally need restarting.

However, database connections in the database connection pool can be validated by running a simple SQL query. If a broken database connection is detected in the pool, a new one is created to replace it.

To do this, you need to specify an optional `<validation-query/>` element (in the `dbconfig.xml` file of your [Jira home directory](#)), whose content is the query which validates connections in the database connection pool. See the following procedure for details.

## Ensuring Jiravalidates connections to its database

1. Shut down Jira(or the Tomcat installation running Jira).
2. Edit the `dbconfig.xml` file at the root of your [Jira home directory](#)or use the **Advanced** tab of the [Jira configuration tool](#)to configure the relevant settings.
3. Configure the validation query for your type of database:
  - If editing the `dbconfig.xml` file, add the `<validation-query/>` element with the appropriate validation query for your type of database, as shown in the example below for MySQL. (See [Determining the validation query](#) below for details.)

```
<?xml version="1.0" encoding="UTF-8"?>

<jira-database-config>
  <name>defaultDS</name>
  <delegator-name>default</delegator-name>
  <database-type>mysql</database-type>
  <jdbc-datasource>
    <url>jdbc:mysql://dbserver:3306/jiradb?useUnicode=true&characterEncoding=UTF8&
sessionVariables=storage_engine=InnoDB</url>
    <driver-class>com.mysql.jdbc.Driver</driver-class>
    <username>jiradbuser</username>
    <password>password</password>
    <pool-min-size>20</pool-min-size>
    <pool-max-size>20</pool-max-size>
    <pool-max-wait>30000</pool-max-wait>

    <validation-query>select 1</validation-query>
    <min-evictable-idle-time-millis>60000</min-evictable-idle-time-millis>
    <time-between-eviction-runs-millis>300000</time-between-eviction-runs-millis>

    <pool-max-idle>20</pool-max-idle>
    <pool-remove-abandoned>true</pool-remove-abandoned>
    <pool-remove-abandoned-timeout>300</pool-remove-abandoned-timeout>

    <pool-test-while-idle>true</pool-test-while-idle>
    <pool-test-on-borrow>false</pool-test-on-borrow>
    <validation-query-timeout>3</validation-query-timeout>

  </jdbc-datasource>
</jira-database-config>
```

- If using the [Jira configuration tool](#), on the **Advanced** tab, select the **Validation Query** checkbox and enter the appropriate validation query for your type of database. (See [Determining the validation query](#) below for details.)
4. Specify a validation query timeout for your validation query, whose value is the appropriate length of time (in seconds) that the system should wait for a validation query to succeed before the system considers the database connection broken:
  - If editing the `dbconfig.xml` file, add the `<validation-query-timeout/>` element with the appropriate length of time (in seconds). ⚠ This should **only** be done for MySQL.
  - If using the [Jira configuration tool](#), on the **Advanced** tab, select the **Validation Query Timeout** checkbox and enter the appropriate length of time (in seconds).
5. You may wish to specify the following options, which relate to the above validation query options (see [Tuning database connections - connection pool settings section](#)for details):

JIRA configuration tool 'Advanced' tab option	Element in <code>dbconfig.xml</code>
Test While Idle	<code>pool-test-while-idle</code>
Time Between Eviction Runs	<code>time-between-eviction-runs-millis</code>
Minimum Evictable Idle Time	<code>min-evictable-idle-time-millis</code>

6. Save your edited `dbconfig.xml` file (or click the **Save** button if using the [Jira configuration tool](#)).
7. Restart JIRA (or the Tomcat installation running Jira).

**ⓘ Please Note:** If you continue to have problems with connections closing, you may need to set the `time-between-eviction-runs-millis` parameter to a lower value or as a last resort, set `test-on-borrow` to `true`. For more information about `test-on-borrow`, see [Tuning database connections - connection pool settings section](#).

### Determining the validation query and timeout

Different database types have slightly different SQL syntax requirements for their validation query. The validation query should be as simple as possible, as this is run every time a connection is retrieved from the pool. The validation query timeout should only be set for MySQL.

The following validation queries are recommended for the following types of databases:

Database type	Validation query	Validation query timeout
MySQL	<code>select 1</code>	3
Microsoft SQL Server	<code>select 1</code>	N/A
Oracle	<code>select 1 from dual</code>	N/A
PostgreSQL	<code>select version();</code>	N/A

**⚠** If the **Validation query timeout** is used on any database other than MySQL it will cause significant problems with the Jira instance.

### Result

You should now be able to recover from a complete loss of all connections in the database connection pool without the need to restart Jira or the application server running JIRA.

### ⚠ Performance considerations:

- Setting this option has a performance impact. The overall decrease in performance should be minimal, as the query itself is quick to run. In addition, the query will only execute when you make a connection. Thus, if the connection is kept for the duration of a request, the query will only occur once per request.
- If you are running a large Jira installation, you may wish to assess the performance impact of this change before implementing it.

# Encrypting database password

To add extra security to your Jira instance, you can encrypt the database password that is stored in the configuration file used by Jira to access your database. We've prepared different encryption methods for basic and advanced users. Additionally, you can create your own encryption based on our Cipher interface.



This solution is an obfuscation, which doesn't assure real security. Jira still needs to use the plain text password to connect to your database, so the configuration will contain all the information needed to decrypt the password. An attacker could act like Jira to obtain the password. We recommend that you secure the server where Jira and the database reside.

## Basic encryption

This method uses a Base64 cipher, which is a simple obfuscation. It's recommended for users who don't want to store passwords in plain text, or have to meet specific requirements to encode them.

[Learn more](#)

## Advanced encryption

This method uses Algorithm cipher that allows you to choose the algorithm used to encrypt your password. It provides more security as you don't have to store the encrypted password anywhere in the configuration file, which makes it difficult to find and decrypt.

[Learn more](#)

## Creating your own Cipher

If you have extra requirements for storing the password, you can create your own Cipher based on our implementation and examples. To do this, you will need Java knowledge and some basic knowledge of Maven.

[Learn more](#)

# Basic encryption

In this method, we'll use Base64Cipher, which is a simple obfuscation.

## Step 1. Encrypting the password

1. Go to `<Jira-installation-directory>/bin`.
2. Run the following command to encrypt your password. Additionally, you can use optional arguments described below.

```
java -cp ".*" com.atlassian.db.config.password.tools.CipherTool
```

- **-- silent -s**: limits logging to minimum
- **-- help -h**: prints a help message with all parameters
- **-- mode -m**: defines what to do with the password, either **encrypt** or **decrypt**. If omitted, 'encrypt' will be used.
- **-- password -p**: plain text password. If omitted, you'll be asked to enter it. We recommend that you omit this parameter so your password is not stored in the history.

```
main DEBUG [db.config.password.DefaultCipherProvider] Initiate cipher provider class: com.atlassian.db.config.password.ciphers.base64.Base64Cipher
main DEBUG [password.ciphers.base64.Base64Cipher] Initiate Base64Cipher
main DEBUG [password.ciphers.base64.Base64Cipher] Encrypting data...
main DEBUG [password.ciphers.base64.Base64Cipher] Encryption done. Success! Put those lines in database config xml file:
<atlassian-password-cipher-provider>com.atlassian.db.config.password.ciphers.base64.Base64Cipher<
/atlassian-password-cipher-provider> <password>ZGZnZGZn</password> and restart the instance.
```

## Step 2. Adding the encrypted password to dbconfig.xml

1. Go to Jira home directory and back up the `dbconfig.xml` file. Move the backup to a safe place outside of your Jira server.
2. Edit the `dbconfig.xml` file.
3. Replace the `<password>` tag with the tags generated by CLI. For example:

```
<atlassian-password-cipher-provider>com.atlassian.db.config.password.ciphers.base64.Base64Cipher<
/atlassian-password-cipher-provider> <password>put-your-obfuscated-password-here</password>
```

4. Restart Jira.

## Decrypting the password

To decrypt the password, extend the command with the **-m decrypt** parameter:

```
java -cp ".*" com.atlassian.db.config.password.tools.CipherTool -m decrypt
```

When asked for a password, provide the encrypted one from your `dbconfig.xml` file.

## Troubleshooting

To revert the changes, remove the `<atlassian-password-cipher-provider>` tag from the `dbconfig.xml` file, and change the encrypted password to a plain text one.

The setup screen means that Jira couldn't connect to the database to access your configuration, most probably because of an error with decrypting your password.

To solve this problem, open `<Jira_home_directory>/log/atlassian-jira.log`, and check the lines after: Reading database configuration from.

You'll probably see the following message:

```
[c.a.j.config.database.DatabaseConfigHandler] Trying to get encrypted password from xml and decrypt it
[c.a.d.c.p.ciphers.base64.Base64Cipher] Runtime Exception thrown when decrypting:
```

If that's the case, read the message, as it contains details about the error and a possible solution. If the error is `java.lang.IllegalArgumentException`, you will need to encrypt the password again.

To investigate this problem, open `<Jira_home_directory>/log/atlassian-Jira.log`, and check the lines after: Reading database configuration from.

You'll probably see the following messages:

```
[c.a.j.config.database.DatabaseConfigHandler] Trying to get encrypted password from xml and decrypt it
[c.a.j.config.database.DatabaseConfigHandler] Database password decryption success!
[c.a.config.bootstrap.DefaultAtlassianBootstrapManager] Could not successfully test your database:

[c.a.jira.health.HealthChecks] JIRA couldn't connect to your database
[c.a.jira.health.HealthChecks] JIRA failed to establish a connection to your database.
```

This means that Jira decrypted the password successfully, but the password itself is incorrect. You can verify that by completing these steps:

1. Open the `dbconfig.xml` file, and copy the encrypted password.
2. [Decrypt the password](#).
3. Check if the decrypted password is the same as the one in your backup `dbconfig.xml` file.

# Advanced encryption

In this method, we'll use `AlgorithmCipher` that allows you to choose the algorithm used to encrypt your password.

## Before you begin: Preparing the JSON object

You will provide all arguments required to encrypt your password in a JSON object. Prepare it beforehand by using the information and examples below.

Field	Description
<code>plainTextPassword</code>	Password in plain text.
<code>algorithm</code>	You can choose one of the following algorithms: <ul style="list-style-type: none"><li>• AES/CBC/PKCS5Padding</li><li>• DES/CBC/PKCS5Padding</li><li>• DESede/CBC/PKCS5Padding</li></ul>
<code>algorithmKey</code>	The algorithm key must correspond with the algorithm chosen above: <ul style="list-style-type: none"><li>• AES</li><li>• DES</li><li>• DESede</li></ul>

```
{"plainTextPassword": "yourPassword", "algorithm": "AES/CBC/PKCS5PADDING", "algorithmKey": "AES" }
```

## Step 1: Encrypting the password

1. Go to `<Jira-installation-directory>/bin`.
2. Run the following command to encrypt your password. You can also use optional parameters described below.

```
java -cp ".*" com.atlassian.db.config.password.tools.CipherTool -c com.atlassian.db.config.password.ciphers.algorithm.AlgorithmCipher
```

- **-- silent -s**: limits logging to minimum
- **-- help -h**: prints a help message with all parameters
- **-- mode -m**: defines what to do with the password, either **encrypt** or **decrypt**. If omitted, 'encrypt' will be used.
- **-- password -p**: JSON object with required arguments. If omitted, you'll be asked to enter it. We recommend that you omit this parameter so your password is not stored in the history.

3. After running the command, you'll be asked to provide the required arguments in a JSON object. Prepare it based on the information from [Before you begin](#).

The encryption tool generated three files that were used to encrypt your password, and printed the output JSON object that you'll later add to the `dbconfig.xml` file.

## Step 2: Securing the generated files

Move the files generated by the tool to a secure place, and change them to read-only. Jira needs to be able to access and read those files to decrypt your password and connect to the database.

The following files have been generated:

- `javax.crypto.SealedObject_[timestamp]`  
File with the encrypted password.

- `javax.crypto.spec.SecretKeySpec_[timestamp]`  
Key used to encrypt your password. You will need this file to decrypt your password.
- `java.security.AlgorithmParameters_[timestamp]`  
Algorithm parameters used to encrypt your password. You will need this file only if you wanted to [recreate an encrypted password](#).

### Step 3: (Optional) Storing file paths as environment variables

 This step is optional, but we recommend that you do it for extra security.

You can store paths to the generated files as environment variables. If the paths aren't present in the `dbconfig.xml` file, Jira will automatically look for them in the specific environment variables. In this way, file paths will not be stored in the `dbconfig.xml` file, making it difficult to locate the files used for encryption.

1. Store the two of the generated files as environment variables. You don't need to add the file with algorithm parameters, because Jira will not be using it to decrypt the password. You must use the following environment variables:

```
com_atlassian_db_config_password_ciphers_algorithm_javax_crypto_spec_SecretKeySpec
com_atlassian_db_config_password_ciphers_algorithm_javax_crypto_SealedObject
```

For example:

```
export com_atlassian_db_config_password_ciphers_algorithm_javax_crypto_spec_SecretKeySpec=/home
/jira/javax.crypto.spec.SecretKeySpec_123456789
export com_atlassian_db_config_password_ciphers_algorithm_javax_crypto_SealedObject=/home/jira
/javax.crypto.SealedObject_123456789
```

2. Edit the output from [Step 1: Encrypting the password](#) and remove paths to the files. The final output should look similar to the following JSON object:

```
<atlassian-password-cipher-provider>com.atlassian.db.config.password.ciphers.algorithm.
AlgorithmCipher</atlassian-password-cipher-provider>
<password>{</password>
```

### Step 4: Adding the encrypted password to dbconfig.xml

Add the output JSON object to the `dbconfig.xml` file, replacing your current password.

1. Go to Jira home directory and back up the `dbconfig.xml` file. Move the backup to a safe place outside of your Jira server.
2. Edit the `dbconfig.xml` file.
3. Replace the `<password>` tag with the output JSON object. Depending on whether you used environment variables or not, adjust the JSON object to one of the following examples:
  - If you stored file paths as environment variables, remove the paths from the output. It should look like the following example:

```
<atlassian-password-cipher-provider>com.atlassian.db.config.password.ciphers.algorithm.
AlgorithmCipher</atlassian-password-cipher-provider>
<password>{</password>
```

- If you didn't use environment variables and want to stick to file paths in the `dbconfig.xml` file, make sure you updated them after moving the files to a secure place. The output should look like the following example:

```
<atlassian-password-cipher-provider>com.atlassian.db.config.password.ciphers.algorithm.
AlgorithmCipher</atlassian-password-cipher-provider>
<password>{"sealedObjectFilePath":"/home/jira/javax.crypto.SealedObject_123456789",
"keyFilePath":"/home/jira/javax.crypto.spec.SecretKeySpec_123456789"}</password>
```

- **WINDOWS** You need to additionally escape the file paths and change double quotes (") surrounding the path to single quotes (') to avoid JSON parsing errors. The paths should look like the following example:

```
<atlassian-password-cipher-provider>com.atlassian.db.config.password.ciphers.algorithm.
AlgorithmCipher</atlassian-password-cipher-provider>
<password>{"sealedObjectFilePath":"'C:\\jira\\javax.crypto.SealedObject_123456789',
"keyFilePath":"'C:\\jira\\javax.crypto.spec.SecretKeySpec_123456789'"}</password>
```

#### 4. Restart Jira.

## Decrypting the password

To decrypt the password, extend the command with the **-m decrypt** parameter:

```
java -cp ".*" com.atlassian.db.config.password.tools.CipherTool -c com.atlassian.db.config.password.
ciphers.algorithm.AlgorithmCipher -m decrypt
```

When asked for the JSON object, provide the one from your `dbconfig.xml` file.

#### Sample JSON object when using file paths

```
{"sealedObjectFilePath":"/home/jira/javax.crypto.SealedObject_123456789",
"keyFilePath":"/home/jira/javax.
crypto.spec.SecretKeySpec_123456789"}
```

#### Sample JSON object when using environment variables

```
{}
```

## Recreating an encrypted password

When you lose the encrypted password and encrypt the plain text password once again, the new encrypted password will look differently. This is not an issue, as it will still represent the same plain text password. However, in some cases, you might want to keep the consistency, for example by having the same encrypted password for all Jira Data Center nodes.

To encrypt the password in the exact same way as you did before, you will need the key used to encrypt the original password and the algorithm parameters. Both of these were generated by the encryption tool and saved in the following files:

- **Key:** `javax.crypto.spec.SecretKeySpec_[timestamp]`
- **Algorithm parameters:** `java.security.AlgorithmParameters_[timestamp]`

Once you've located these files, you can point the encryption tool to their location by using two extra fields in the JSON object. Below you can find the description of these fields and a sample JSON object.

Field	Description
<code>keyFilePath</code>	Path to a file that contains the key used to encrypt your original password, e.g. <code>javax.crypto.spec.SecretKeySpec_[timestamp]</code> .  If you stored the file path as environment variable, you can omit this parameter.

algorithmParametersFilePath	Path to a file that contains the algorithm parameters used to encrypt your original password, e.g. <code>java.security.AlgorithmParameters_[timestamp]</code> .
<pre>{ "plainTextPassword": "yourPassword", "algorithm": "AES/CBC/PKCS5PADDING", "algorithmKey": "AES",   "algorithmParametersFilePath": "java.security.AlgorithmParameters_123456789", "keyFilePath": "javax.crypto.spec.SecretKeySpec_123456789" }</pre>	

To encrypt the password, follow the steps in [Step 1: Encrypting the password](#), and use the JSON object with the key and algorithm parameters.

### Troubleshooting

To revert the changes, remove the `<atlassian-password-cipher-provider>` tag from the `dbconfig.xml` file, and change the encrypted password to a plain text one.

The setup screen means that Jira couldn't connect to the database to access your configuration, most probably because of an error with decrypting your password.

To solve this problem, open `<Jira_home_directory>/log/atlassian-jira.log`, and check the lines after: Reading database configuration from.

You'll probably see the following message:

```
[c.a.j.config.database.DatabaseConfigHandler] Trying to get encrypted password from xml and decrypt it
[c.a.d.c.p.ciphers.algorithm.AlgorithmCipher] Runtime Exception thrown when decrypting:
```

If that's the case, read the message, as it contains details about the error and a possible solution.

- If the error is related to missing files, there might be a problem with your environment variables. They could have been deleted, or are no longer available if you changed the environment from staging to production. To verify that, try adding file paths to the JSON object in the `dbconfig.xml` file.
- If you're seeing some Bouncy Castle errors, you will need to encrypt the password again.

To investigate this problem, open `<Jira_home_directory>/log/atlassian-jira.log`, and check the lines after: Reading database configuration from.

You'll probably see the following messages:

```
[c.a.j.config.database.DatabaseConfigHandler] Trying to get encrypted password from xml and decrypt it
[c.a.j.config.database.DatabaseConfigHandler] Database password decryption success!
[c.a.config.bootstrap.DefaultAtlassianBootstrapManager] Could not successfully test your database:

[c.a.jira.health.HealthChecks] JIRA couldn't connect to your database
[c.a.jira.health.HealthChecks] JIRA failed to establish a connection to your database.
```

This means that Jira decrypted the password successfully, but the password itself is incorrect. You can verify that by completing these steps:

1. Open the `dbconfig.xml` file, and copy the encrypted password.
2. [Decrypt the password](#).
3. Check if the decrypted password is the same as the one in your backup `dbconfig.xml` file.

# Creating your own Cipher

You can also create your own Cipher, which might be especially useful if you're required to use a specific vault to store the password.

Pre-requisites:

- Basic knowledge of Maven
- Knowledge of Java

## Step 1: Create a Maven project and get API dependencies

1. Get 'api' and 'base' dependencies.
  - a. Go to `<Jira_installation_directory>/atlassian-jira/WEB-INF/lib`.
  - b. Copy the following jar files:
    - **password-cipher-api-<version>.jar**: This file contains the API.
    - (optional) **password-cipher-base-<version>.jar**: This file contains sample implementation.
2. Create a Maven project.
3. Go to `resources`, and create a new folder `libs`.
4. Copy the jar files to the `libs` folder.
5. You can then use the following `pom`:

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<groupId><your_group_ID></groupId>
<artifactId><your_artifact_ID></artifactId>
<version><your_version></version>

<properties>
<maven.compiler.source>1.8</maven.compiler.source>
<maven.compiler.target>1.8</maven.compiler.target>
</properties>

<repositories>
<repository>
<id>local-maven-repo</id>
<url>file:///${project.basedir}/libs</url>
</repository>
</repositories>
<build>
<resources>
<resource>
<directory>src/main/resources/libs</directory>
<excludes>
<exclude>*</exclude>
</excludes>
<filtering>>false</filtering>
</resource>
</resources>
</build>

<dependencies>
<dependency>
<groupId>com.atlassian.db.config</groupId>
<artifactId>password-cipher-api</artifactId>
<version><api_version></version>
<scope>provided</scope>
</dependency>
<dependency>
<groupId>com.atlassian.db.config</groupId>
<artifactId>password-cipher-base</artifactId>
<version><base_version></version>
<scope>provided</scope>
</dependency>
</dependencies>

</project>

```

## Step 2: Implement the Cipher interface

The Cipher interface contains only two methods `encrypt` and `decrypt`. `Decrypt` will be called during Jira startup, which means that long running tasks can affect the startup time. `Encrypt` will not be called by Jira, as it's only used in the encryption tool.

You can use `Base64Cipher` and `AlgorithmCipher` as examples.

## Step 3: Test your implementation

The encryption tool, described in [Basic encryption](#) and [Advanced encryption](#), uses the same code as Jira to decrypt the password. You can use it to test your implementation.

Assuming that CLI and your jar is in the same folder:

```
java -cp ".*" com.atlassian.db.config.password.tools.CipherTool -c your.package.here.ClassName
```

**Step 4: Make your lib available to Jira**

Jira must be able to access your lib. Your class will be initiated using reflection. Put the lib in the following directory:

```
<Jira_installation_directory>/atlassian-jira/WEB-INF/lib
```

 After upgrading Jira, you'll need to copy your lib to the Jira installation directory again.

# Switching databases

Jira's data can be migrated from one database to:

1. A different database on the same database server,
2. The same database type on a different server (e.g. from one PostgreSQL server to another PostgreSQL server) or
3. A different type of database server (e.g. from a MySQL server to a PostgreSQL server).

⚠ For migrating Jira to another server, please refer to the [Migrating Jira to another server](#) document instead.

**To do this, follow the appropriate procedure:**

- [Migrating Jira's data to the same type of database](#) (covers scenarios 1 and 2 above)
- [Migrating Jira's data to a different type of database server](#) (covers scenario 3 above)

If you are planning to migrate to an Amazon Aurora database, you can also review their documentation for additional advice. See [Additional AWS resources for Amazon Aurora](#) below.

## Known issues

- [Database migration to SQL Server fails because of duplicate entries](#)

## Migrating Jira's data to the same type of database

Use this procedure to migrate Jira's data to:

- A different database on the same database server, or
  - The same database type on a different database server (e.g. from one PostgreSQL server to another PostgreSQL server).
1. Use your database server's native tools to either:
    - Copy your Jira database to a new database on the same database server installation, or
    - Copy/migrate your Jira database to a new database of the same type on a different database server installation.
- ⓘ Please Note:**
- If you are unable to do either of these tasks, use the [Migrating Jira's database to a different type of database server](#) procedure (below) instead.
  - You could use this procedure to migrate Jira's data to a different type of database server (e.g. MySQL to PostgreSQL). However, you would need to find tools that support these processes. Furthermore, Atlassian does not provide support for this strategy.
2. Once your new database has been populated with Jira's data, shut down your Jira server.
  3. Make a backup of your [Jira home directory](#) and [Jira installation directory](#).
  4. Reconfigure your Jira server's connection to your database:
    - If you installed a 'Recommended' distribution of Jira, you can use the [Jira configuration tool](#) (by running `bin/config.sh` (for Linux/Solaris) or `bin\config.bat` (for Windows) in your [Jira installation directory](#)), which provides a convenient GUI that allows you to reconfigure Jira's database connection settings.
    - If any of the following points applies to your situation, you need to manually configure the `dbconfig.xml` file in your [Jira home directory](#). Refer to the appropriate database configuration guide in the [Connecting Jira to a database](#) section for the manual configuration instructions.
      - You have a console-only connection to your Jira server
      - You would prefer to configure your database connection manually (for custom configuration purposes).

## Migrating Jira's data to a different type of database server

Use this procedure to migrate Jira's data to a different type of database server (e.g. from a MySQL server to a PostgreSQL server).

✔ You can also use this procedure if your Jira installation is currently using the internal H2 database (which is only supported for evaluating Jira) and you need to switch your Jira installation across to using a [supported database](#) (which are supported for Jira installations used in a production environment).

1. Create an export of your data as an XML backup. See [Backing up data](#) for details.
2. Create a new database on your new database server to house Jira's data. See the appropriate database configuration guide in the [Connecting JIRA to a database](#) section for the database creation instructions.
3. Shut down your Jira server.
4. Make a backup of your [Jira home directory](#) and [Jira installation directory](#).
5. Delete the `dbconfig.xml` file in your [Jira home directory](#).
6. Restart Jira and you should see the first step of the [JIRA setup wizard](#) for configuring your database connection.
7. Configure Jira's connection to your new database (created in step 2 above) and click the **'Next'** button.
8. On the 'Application Properties' setup page, click the **'import your existing data'** link and [restore your data](#) from the XML backup created in step 1 above.

## Additional AWS resources for Amazon Aurora

AWS has some helpful guides for setting up an Aurora database and migrating to it:

- [Modular Architecture for Amazon Aurora PostgreSQL](#): a Quick Start that guides you through the deployment of a PostgreSQL-compatible Aurora Database cluster. This cluster is similar to the one set up by the [AWS Quick Start for Jira](#) (one writer and two readers, preferably in different availability zones).
- [Upgrading the PostgreSQL DB Engine for Amazon RDS](#): shows you how to upgrade your database engine to a supported version before migrating it to Amazon Aurora.
- [Migrating Data to Amazon Aurora PostgreSQL](#): contains instructions for migrating from Amazon RDS to a PostgreSQL-compatible Amazon Aurora cluster.
- [Best Practices with Amazon Aurora PostgreSQL](#): contains additional information about best practices and options for migrating data to a PostgreSQL-compatible Amazon Aurora cluster.

Amazon also offers an [AWS Database Migration Service](#) to facilitate a managed migration. This service offers minimal downtime, and supports migrations to Aurora from a wide variety of source databases.

# Connecting Jira applications to SQL Server 2019

These instructions will help you connect Jira to a Microsoft SQL Server 2019 database.

Before you begin

- If you're [Migrating Jira applications to another server](#), create an export of your data as an [XML backup](#). You will then be able to transfer data from your old database to your new database, as described in [Switching databases](#).
- Stop Jira before you begin, unless you just started the installation and are running the Setup Wizard.

## 1. Create and configure the SQL Server database

**i** When creating the database, remember your **database name**, **user name**, **schema name**, and **port number**, because you'll need them later to connect Jira to your database.

### 1. Create a database for Jira (e.g. `jiradb`).

- Make sure the collation type is **case-insensitive**.

We support `SQL_Latin1_General_CP437_CI_AI` and `Latin1_General_CI_AI` as case-insensitive, accent-insensitive, and language neutral collation types. If your SQL Server installation's collation type settings have not been changed from their defaults, check the collation type settings.

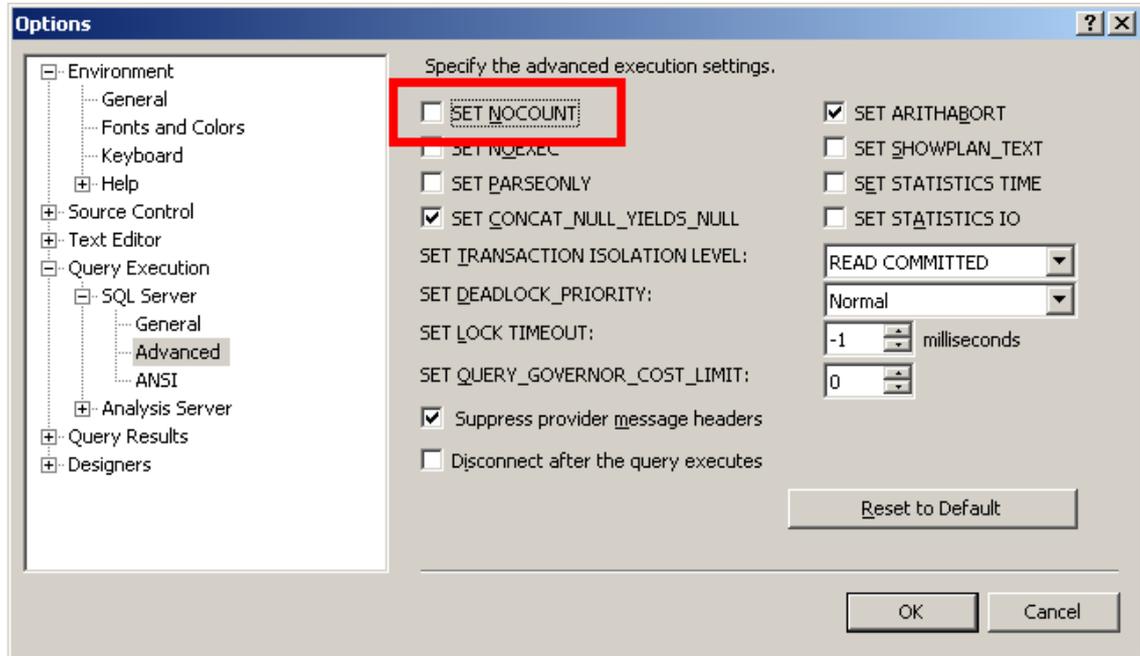
- SQL Server uses Unicode encoding to store characters. This is sufficient to prevent any possible encoding problems.
2. Create a database user which Jira will connect as (e.g. `jiradbuser`). This user should *not* be the database owner, but *should* be in the `db_owner` role.
  3. Create an empty 'schema' in the database for the Jira tables (e.g. `jiraschema`).

A 'schema' in SQL Server 2019 is a distinct namespace used to contain objects and is *different* from a traditional database schema. You are not required to create any of Jira's tables, fields or relationships (Jira will create these objects in your empty schema when it starts for the first time). You can read more on SQL Server 2019 schemas in the relevant [Microsoft documentation](#).

4. Make sure that the database user has permission to connect to the database, and to create and populate tables in the newly-created schema.
5. Make sure that TCP/IP is enabled on SQL Server and is listening on the correct port. A default SQL Server installation uses port number 1433.
6. Make sure that SQL Server is operating in the appropriate authentication mode.

By default, SQL Server operates in 'Windows Authentication Mode'. However, if your user is not associated with a trusted SQL connection, i.e. 'Microsoft SQL Server, Error: 18452' is received during Jira startup, you will need to change the authentication mode to 'Mixed Authentication Mode'. Read the Microsoft documentation on authentication modes and changing the authentication mode to 'Mixed Authentication Mode'

7. Turn off the SET NOCOUNT option.
  - a. Open SQL Server Management Studio.
  - b. Go to **Tools > Options > Query Execution > SQL Server > Advanced**, and clear the **SET NOCOUNT** check box.



- c. Right-click your server in the Object Explorer, and go to **Properties > Connections > Default Connections**. Clear the **no count** option.
8. Access the **Query Console** by right clicking on the newly created database and selecting 'New Query'. Run the following command to set the isolation level.

```
ALTER DATABASE THE-NEW-DATABASE-CREATED-FOR-JIRA SET READ_COMMITTED_SNAPSHOT ON
```

## 2. Configure Jira to connect to the database

There are two ways to configure your Jira server to connect to your SQL Server database.

- **Using the Jira setup wizard** Use this method, if you have just installed Jira and are setting it up for the first time. Your settings will be saved to the `dbconfig.xml` file in your [Jira application home directory](#).

The [Jira setup wizard](#) will display when you access Jira for the first time in your browser.

1. In the first screen, 'Configure Language and Database', set **Database Connection to My own database**.
  2. Set **Database Type to SQL Server**.
  3. Fill out the fields, as described in the [Database connection fields](#) section below.
  4. Test your connection and save.
- **Using the Jira configuration tool** Use this method, if you have an existing Jira instance. Your settings will be saved to the `dbconfig.xml` file in your [Jira application home directory](#).
    1. Run the Jira configuration tool as follows:
      - **Windows:** Open a command prompt and run `config.bat` in the `binsub`-directory of the [Jira installation directory](#).
      - **Linux/Unix:** Open a console and execute `config.sh` in the `binsub`-directory of the [Jira installation directory](#).
        - This may fail with the error as described in our [Unable to Start Jira applications Config Tool due to No X11 DISPLAY variable was set error](#) KB article. Please refer to it for the workaround.
    2. Navigate to the **Database** tab and set **Database type to SQL Server**.
    3. Fill out the fields, as described in the [Database connection fields](#) section below.
    4. Test your connection and save.
    5. Restart Jira.

### Database connection fields

The table shows the fields you'll need to fill out when connecting Jira to your database. You can also refer to them, and the `sampledbconfig.xml` file below, if you'd like to create or edit the `dbconfig.xml` file manually.

Setup wizard / Configuration tool	dbconfig.xml	Description
<b>Hostname</b>	Located in the <code>&lt;url&gt;</code> tag (bold text in example below): <code>&lt;url&gt;jdbc:sqlserver://<b>dbserver</b>:1433;databaseName=jiradb&lt;/url&gt;</code>	The name or IP address of the machine that the SQL Server server is installed on.
<b>Port</b>	Located in the <code>&lt;url&gt;</code> tag (bold text in example below): <code>&lt;url&gt;jdbc:sqlserver://dbserver :<b>1433</b>;databaseName=jiradb&lt;/url&gt;</code>	The TCP/IP port that the SQL Server server is listening on. You can leave this blank to use the default port.
<b>Database</b>	Located in the <code>&lt;url&gt;</code> tag (bold text in example below): <code>&lt;url&gt;jdbc:sqlserver://dbserver :1433;databaseName=<b>jiradb</b>&lt;/url&gt;</code>	The name of your SQL Server database (into which Jira will save its data). You should have created this in Step 1 above.
<b>Username</b>	Located in the <code>&lt;username&gt;</code> tag (see bold text in example below): <code>&lt;username&gt; <b>jiradbuser</b> &lt;/username&gt;</code>	The user that Jira uses to connect to the SQL Server server. You should have created this in Step 1 above.
<b>Password</b>	Located in the <code>&lt;password&gt;</code> tag (see bold text in example below): <code>&lt;password&gt; <b>jiradbuser</b> &lt;/password&gt;</code>	The user's password used to authenticate with the SQL Server server.
<b>Schema</b>	Located in the <code>&lt;schema-name&gt;</code> tag (see bold text in example below): <code>&lt;schema-name&gt; <b>dbo</b> &lt;/schema-name&gt;</code>	The name of the schema that your SQL Server database uses. You should have created this in Step 1 above.

#### Sample dbconfig.xml file

For more information about the child elements of `<jdbc-datasource/>` beginning with `pool` in the `dbconfig.xml` file above, see [Tuning database connections](#).

```
<jira-database-config>
<name>defaultDS</name>
<delegator-name>default</delegator-name>
<database-type>mssql</database-type>
<schema-name>jiraschema</schema-name>
<jdbc-datasource>
  <url>jdbc:sqlserver://dbserver:1433;databaseName=jiradb</url>
  <driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>
  <username>jiradbuser</username>
  <password>password</password>
  <pool-min-size>20</pool-min-size>
  <pool-max-size>20</pool-max-size>
  <pool-max-wait>30000</pool-max-wait>
  <pool-max-idle>20</pool-max-idle>
  <pool-remove-abandoned>true</pool-remove-abandoned>
  <pool-remove-abandoned-timeout>300</pool-remove-abandoned-timeout>

  <validation-query>select 1</validation-query>
  <min-evictable-idle-time-millis>60000</min-evictable-idle-time-millis>
  <time-between-eviction-runs-millis>300000</time-between-eviction-runs-millis>

  <pool-test-while-idle>true</pool-test-while-idle>
  <pool-test-on-borrow>false</pool-test-on-borrow>
</jdbc-datasource>
</jira-database-config>
```

### 3. Start Jira

You should now have Jira configured to connect to your SQL Server database. The next step is to start it up!

# Installing Jira Data Center

These instructions are applicable for installing Jira Software Data Center or Jira Service Management Data Center on your own hardware.



## Other ways to install Jira Data Center

- [Kubernetes](#) Installation on a Kubernetes cluster using our Helm charts
- [AWS](#) hassle-free deployment in AWS using our Quick Start
- [Azure](#) reference templates for Microsoft Azure deployment

## Before you begin

Things you should know about when setting up your Data Center:

See our [Supported platforms](#) page for information on the database, Java, and operating systems you'll be able to use. These requirements are the same for Server and Data Center deployments.

To use Jira Data Center, you must:

- Have a Data Center license (you can [purchase a Data Center license](#) or create an evaluation license at [my.atlassian.com](#))
- Use a [supported](#) external database, operating system and Java version
- Use OAuth authentication if you have [application links](#) to other Atlassian products (such as Confluence)

To run Jira in a cluster, you must also:

- Use a load balancer with session affinity and WebSockets support in front of the Jira cluster. [Load balancer examples](#)
- Have a shared directory accessible to all cluster nodes in the same path (this will be your shared home directory). This must be a separate directory, and not located within the local home or install directory.

In this guide we'll use the following terminology:

- **Installation directory:** The directory where you installed Jira.
- **Local home directory:** The home or data directory stored locally on each cluster node (if Jira is not running in a cluster, this is simply known as the home directory).
- **Shared home directory:** The directory you created that is accessible to all nodes in the cluster via the same path.

## Install Jira Data Center on a single node

If your organization doesn't need high availability or disaster recovery capabilities right now, you can install Jira Data Center without setting up a cluster.

To install Jira Data Center, without setting up a cluster, follow the instructions for Jira Server:

- [Installing Jira applications](#)

The process is almost identical to an ordinary Jira Server installation, just be sure to enter your Data Center license.

## Install Jira Data Center in a cluster

### 1. Install or upgrade your Jira instance

Jira Data Center is available for Jira 7.0, or later. If you're not on this version yet, install or upgrade your Jira instance.

[Jira installation and upgrade guide](#)

### 2. Set up the shared directory

You'll need to create a remote directory that is readable and writable by all nodes in the cluster. There are multiple ways to do this, but the simplest is to use an NFS share.

1. Create a remote directory, accessible by all nodes in the cluster, and name it e.g. `sharedhome`.
2. Stop your Jira instance.
3. Copy the following directories from the Jira local home directory to the new `sharedhome` directory (some of them may be empty).

- `data`
- `plugins`
- `logos`
- `import`
- `export`
- `caches`
- `keys`

### 3. Configure your Jira instance to work in a cluster

1. In the Jira local home directory, create a `cluster.properties` file, with contents as follows:

#### Example `cluster.properties` file:

```
# This ID must be unique across the cluster
jira.node.id = node1
# The location of the shared home directory for all Jira nodes
jira.shared.home = /data/jira/sharedhome
```

For more information and some additional parameters, see [Cluster.properties file parameters](#).

2. **For Linux installations:** We recommend that you increase the maximum number of open files. To do that, add the following line to `<jira-install>/bin/setenv.sh`:

```
ulimit -n 16384
```

3. Start your instance, and [apply the Data Center license](#).

### 4. Add the first node to the load balancer

The load balancer distributes the traffic between the nodes. If a node stops working, the remaining nodes will take over its workload, and your users won't even notice it.

1. Add the first node to the load balancer.
2. Restart the node, and then try opening different pages in Jira. If the load balancer is working properly, you should have no problems with accessing Jira.

### 5. Add the remaining nodes to the cluster

The approach to adding the remaining nodes to the cluster varies with the method that was used to install Jira on the first node (either manually from a `.zip` or `.tar.gz` archive or using a `.bin` or `.exe` installer). Follow the steps that correspond to the original installation method.

1. Copy the Jira installation and home directories from an existing node to the new node.
  2. Ensure the new node can read and write to the shared home directory.
  3. Edit `<home-directory>/cluster.properties` on the new node by providing a unique node ID and an IP address if one was specified.
  4. Start Jira. It will read the configuration from the shared home directory and start without any extra setup.
  5. Take a look around the new Jira instance. Ensure that issue creation, search, attachments, and customizations work as expected.
  6. If everything looks fine, you can configure your load balancer to start routing traffic to the new node. Once you do this, you can make a couple of changes in one Jira instance to see if they're visible in other instances as well. Use the same method to install the same version of Jira on another node in your cluster. During the installation, take note of the locations of the Jira installation and home directory paths.
1. Ensure the new node can read and write to the shared home directory.
  2. Start Jira to allow the application to populate the home directory.
  3. Open Jira in the browser and make sure that you can see the setup page. If the page appears, the installation was successful and you can close the browser.
  4. Stop Jira.
  5. Copy `dbconfig.xml` and `cluster.properties` from the Jira home directory on an existing node to the Jira home directory on the new node.
  6. Copy `server.xml` from `<installation-directory>/conf` on an existing node to `<installation-directory>/conf` on the new node.
  7. Edit `<home-directory>/cluster.properties` on the new node by providing a unique node ID and an IP address if one was specified.
  8. If you modified any [important directories and files](#) (for example, `<installation-directory>/bin/setenv.sh` or `<installation-directory>/conf/web.xml`) on an existing node, copy the modified files to the same locations on the new node.
  9. If Jira runs over SSL, import the SSL certificates to the local Java truststore on the new node to allow Jira to communicate with itself over its base URL.
  10. Start Jira. It will read the configuration from the shared home directory and start without any extra setup.
  11. Take a look around the new Jira instance. Ensure that issue creation, search, attachments, and customizations work as expected.
  12. If everything looks fine, you can configure your load balancer to start routing traffic to the new node. Once you do this, you can make a couple of changes in one Jira instance to see if they're visible in other instances as well.

 While adding your nodes to the cluster, you can check their status by going to  **System > System info**. Your nodes will be listed in the **Cluster nodes** section.

### Cluster.properties file parameters

In addition to the required parameters, the `cluster.properties` file allows you to configure some additional options, mostly related to EhCache.

Parameter	Required	Description/value
<code>jira.node.id</code>	Yes	This unique ID must match the username and the <code>BalancerMember</code> entry in the Apache configuration.
<code>jira.shared.home</code>	Yes	The location of the shared home directory for all Jira nodes.

<code>ehcache.peer.discovery</code>	No	<p>Describes how nodes find each other:</p> <p><code>default</code> Jira will automatically discover nodes (recommended)  <code>automatic</code> Jira will use the EhCache's multicast discovery. This is the historical method used by EhCache, but it can be difficult to configure, and is not recommended by Atlassian.</p> <p>If you set <code>ehcache.peer.discovery=automatic</code> then you need to set the following parameters:</p> <ul style="list-style-type: none"> <li>• <code>ehcache.multicast.address</code></li> <li>• <code>ehcache.multicast.port</code></li> <li>• <code>ehcache.multicast.timeToLive</code></li> <li>• <code>ehcache.multicast.hostName</code></li> </ul> <p>For more info on these parameters, see <a href="#">Ehcache documentation</a>.</p>
<code>ehcache.listener.hostName</code>	No	The hostname of the current node for cache communication. Jira Data Center will resolve this this internally if the parameter isn't set. If you have problems resolving the hostname of the network you can set this parameter.
<code>ehcache.listener.port</code>	No	The port that the node is going to be listening to (default is 40001).  If multiple nodes are on the same host, or if this port is unavailable, you might need to set this parameter manually.
<code>ehcache.object.port</code>	No	The port on which the remote objects bound in the registry receive calls (default is 40011). Make sure you also open this port on your firewall.  If multiple nodes are on the same host, or if this port is unavailable, you might need to set this parameter manually.
<code>ehcache.listener.socketTimeoutMillis</code>	No	By default, this is set to the EhCache default.

## Security

To secure your application, use a firewall or network segregation (or both) to ensure that only permitted nodes connect to Jira Data Center's Ehcache RMI ports. If you use a firewall, you must open ports in the firewall between nodes and the cache, or else you may see cache replication issues. The two default Ehcache RMI ports are 40001 and 40011.

**Not restricting access to the Ehcache RMI ports might compromise your Jira Data Center instance.**

### Monitoring the health of your Data Center

Now that you got your Data Center up and running, we recommend that you keep monitoring its health right from the start. This will help you keep any problems from getting bigger and messing with your work, and you'll always know what's going on in the cluster.

JIRA Data Center is equipped with a set of health checks tools that let you monitor the whole cluster and each node individually, including all important settings. To access the health check tools, go to **Administration** (⚙️) > **System > Support Tools**. All health checks are listed in the **Instance health** tab.

Additionally, you can monitor the health of your remote caches. See [Monitoring the cache replication](#). For more on remote caches, see [Jira Data Center cache replication](#).

In Jira Data Center version 8.6 and later, you can monitor your Data Center cluster. Go to **Jira Administration > System > Clustering** and see how your cluster is doing. For more on cluster monitoring, see [Jira cluster monitoring](#).

# Migrate from Server to Data Center

This page outlines the process for migrating to Jira Data Center.

If you're installing Jira for the first time (you don't have any existing Jira data to migrate), see [Installing Jira Data Center](#).

## Before you begin

Things you should know about when setting up your Data Center:

It's your Jira license that determines the type of Jira you have: Server or Data Center. Jira will auto-detect the license type when you enter your license key, and automatically unlock any license-specific features.

You can [purchase a Data Center license](#) or create an evaluation license at [my.atlassian.com](https://my.atlassian.com)

See our [Supported platforms](#) page for information on the database, Java, and operating systems you'll be able to use. These requirements are the same for Server and Data Center deployments.

Apps extend what your team can do with Atlassian applications, so it's important to make sure that your team can still use their apps after migrating to Data Center. When you switch to Data Center, you'll be required to switch to the Data Center compatible version of your apps, if one is available.

See [Evaluate apps for Data Center migration](#) for more information.

To use Jira Data Center, you must:

- Have a Data Center license (you can [purchase a Data Center license](#) or create an evaluation license at [my.atlassian.com](https://my.atlassian.com))
- Use a [supported](#) external database, operating system and Java version
- Use OAuth authentication if you have [application links](#) to other Atlassian products (such as Confluence)

To run Jira in a cluster, you must also:

- Use a load balancer with session affinity and WebSockets support in front of the Jira cluster. [Load balancer examples](#)
- Have a shared directory accessible to all cluster nodes in the same path (this will be your shared home directory). This must be a separate directory, and not located within the local home or install directory.

## Migrate to non-clustered Data Center

### Review and upgrade your apps

If you have any apps installed on your site, you'll need to upgrade to the Data Center app version, if one is available. To avoid any impact to your apps, we recommend you do this before you enter your Jira Data Center license key. Learn more about [upgrading server apps when you migrate to Data Center](#).

### Upgrade your Jira license

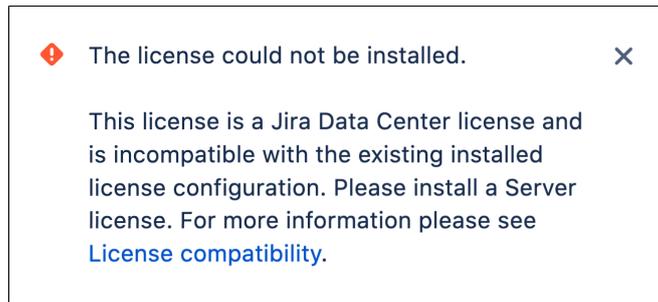
To move from Jira Server to Jira Data Center:

1. Go to **Administration**  > **Applications > Versions and licenses**.
2. Enter your Jira Data Center license key.
3. Restart Jira.

Data Center features such as project and issue archiving, rate limiting, and SAML single sign-on will now be available.

### Can't update the license?

Your Data Center license must match the one you used for Server. For example, if your Server license included Jira Software and Jira Service Management, but the new one has Jira Software only, you won't be able to update it and will most likely see the following error:



If you're looking to just evaluate Data Center or change the products in your license, you can work around it by first switching to an evaluation Server license (for example, Jira Software license, which will remove licenses for other products) and then using the Data Center license.

## Set up your cluster

If your organization requires continuous uptime, scalability, and performance under heavy load, you'll want to run Jira Data Center in a cluster.

To find out more about clustering, including infrastructure requirements, see [Running Jira Data Center in a cluster](#).

If you're ready to set up your cluster now, head to [Set up a Jira Data Center cluster](#).

 Looking to migrate all your Atlassian applications to Data Center? We've got you covered:

- [Migrate to Bitbucket Data Center](#)
- [Migrate to Crowd Data Center](#)
- [Migrate to Confluence Data Center](#)
- [Migrate to Jira Data Center](#)

Considering moving to cloud? [Plan your cloud migration](#).

# Running the setup wizard

The Jira setup wizard allows you to either set up a Jira application for evaluation and demonstration purposes, or for production and testing.

To get started, access your new Jira application in a browser, after you have [installed it](#). Your server will be available at the following URL, if you are using the default port: `http://<jira-server-name>:8080`.

**i** The Jira application setup wizard will only display the first time after you install your Jira application. Once you have completed it, you cannot run it again. However, every setting configured in the setup wizard can be configured via the Jira administration console.

## Evaluation and demonstration

If you want to evaluate or demonstrate a Jira application, let us do most of the set up for you. We will help you set up an Atlassian account if you don't have one, and will generate an evaluation license for you. We'll also set up a H2 database for evaluation purposes (see [Supported Platforms](#)). The only requirement is you have a connection to the internet, as we'll need this to validate and generate Atlassian account details and your evaluation license.

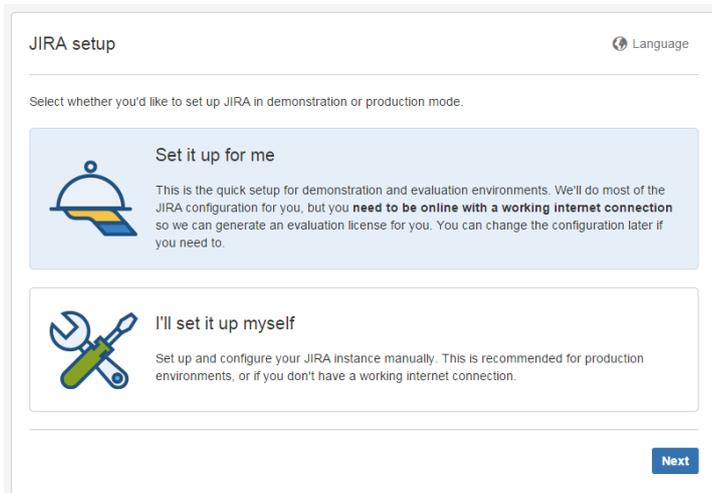
Follow the steps [here](#).

## Production and testing

If you want to set up a Jira application for production or testing purposes before you upgrade, we recommend you follow the custom installation path. This will allow you to connect to your own database if required, and set up your email SMTP server. This path can also be followed if you don't have a connection to the internet. You'll be able to manually paste in a license key.

Follow the steps [here](#).

## Evaluation and demonstration setup



1. Choose the language you would like the Jira application setup and user interface to appear in by selecting the preferred **Language**. Note:

- As soon as you choose a language from the **Language** drop-down list, the Jira application user interface will switch to that language.
- Be aware that some languages may have more comprehensive translations than others.

2. Select **Set it up for me** and click **Next**.

3. Enter your Atlassian ID email address, or if you don't have an Atlassian ID account, enter an email address you'd like to use and have access to, and click **Next**.

4. We'll validate your account, and create a new one if needed. Remember the details you use for your Atlassian ID account, as these will be the same credentials for your Jira system administrator. You can change the system administrator details within Jira when you're up and running. Check out [Managing Global Permissions](#) and [Managing Users](#) for more information. Select **Next** to generate your license.

5. Select **Next** to finish the setup process. This may take a minute or two.

6. Once the setup is complete, you're ready to get started! Select **Launch Jira** to get going!

✔ That's it!

## Production and testing setup

1. Choose the language you would like the Jira application user interface to appear in by selecting the preferred **Language**. Note:

- As soon as you choose a language from the **Language** dropdown list, the Jira application user interface will switch to that language.
- Be aware that some languages may have more comprehensive translations than others.

2. Select **I'll set it up myself** and click **Next**.

3. Configure a database for Jira.

Choose between connecting Jira to the bundled database or your own database.

Database Connection	Recommended for	Instructions	Notes
Bundled database	Evaluations only	Go to the next step. The bundled H2 database will be automatically configured by the setup wizard.	<ul style="list-style-type: none"> <li>• The H2 database is suitable for evaluation and demonstration purposes only. We recommend connecting to a <a href="#">supported database</a> for production environments.</li> </ul>

Your own database	Production use	<ol style="list-style-type: none"> <li>1. Choose a database. See our list of <a href="#">supported databases</a> first.</li> <li>2. Configure the database connection. If you need help, see the guides on <a href="#">Connecting Jira to a database</a>. Note, the fields displayed on this screen are identical to those on the <a href="#">Jira configuration tool</a>.</li> </ol>	<ul style="list-style-type: none"> <li>• Your external database must be a newly-created (or empty) database.</li> <li>• Database connection pool You cannot configure your database connection pool size through the setup wizard. You can do this subsequently using the <a href="#">Jira configuration tool</a> or manually (described on each specific <a href="#">database configuration guide</a>).</li> <li>• MySQL database The MySQL driver is not bundled with Jira (see <a href="#">Connecting Jira applications to MySQL 5.7</a>). You need to copy the driver into the lib folder of your Jira installation and restart Jira/Jira service before completing the setup wizard.</li> </ul>
-------------------	----------------	---	--

4. If you're connecting to your own database, click **Test connection** to make sure Jira can connect. Click **Next** when you're ready to proceed.

5. You need to configure the Title, Mode, and Base URL for your instance:

Setting	Instructions	Notes
Application Title	Choose a title that helps identify your installation and its purpose.	<ul style="list-style-type: none"> <li>• The application title will be displayed on the login page and the dashboard.</li> <li>• After you have completed the setup wizard, you may also want to configure the logo and color scheme of your installation.</li> </ul>
Mode	Choose a mode that suits how you use your issue tracker.	<ul style="list-style-type: none"> <li>• Setting the mode to public enables public signup. Note, that allowing anyone to sign up can cause you to <b>exceed the user limit on your Jira application license</b>.</li> <li>• A public issue tracker can be useful for gathering feedback and bug reports directly from customers. A private issue tracker may be more suitable for tracking the development progress of your team.</li> </ul>
Base URL	Specify the base URL that users will use to access your instance.	<ul style="list-style-type: none"> <li>• You can only configure Jira to respond to a single URL and <b>this setting must match the URL that your users request for accessing your Jira instance</b>. You cannot (for example) have a different hostname or URL for internal and external users. Any mismatch between this Base URL setting and the URL requested by your Jira application users will cause problems with dashboard gadgets.</li> <li>• This URL is also used in outgoing email notifications as the prefix for links to Jira issues.</li> </ul>

#### Further information:

- If you need to change these settings after setting up your application, you can configure them via the Jira administration console. For details, see [Configuring Jira options](#).
- Jira will store your automated backups, file attachments and indexes in your [Jira home directory](#).

Click **Next** when you've configured all the application properties to your liking.

6. You are required to enter a Jira application license key before you can use your application. If you don't have a Jira application license key, you can get the setup wizard to create an evaluation license for you. Evaluation license keys will allow you to use a fully functional installation for 30 days.

License keys for Atlassian applications are linked to your account at [my.atlassian.com](https://my.atlassian.com). If you don't have any [my.atlassian.com](https://my.atlassian.com) account, you can create one and get the setup wizard to create an evaluation license for you.

7. Enter the details for the administrator account for the installation. The account will be granted the [Jira system administrator permission](#).

You can create additional Jira system administrator and Jira administrator accounts after you have set up Jira. Click **Next** when you've entered the details.

8. Set up your email SMTP server. This step is optional. You can configure email notifications after you have set up Jira if you wish.

If you want to configure email notifications at this stage, you will need to set up a connection to a mail server. See this page for further instructions: [Configuring Jira's SMTP Mail Server to Send Notifications](#). Click **Finish** to complete the setup.

 **Congratulations, you have completed setting up your new Jira application installation!**

Detailed information on using and administering Jira and your Jira applications can be found in the rest of the Administering Jira applications documentation.

# Licensing and application access

To grant users log in access to a Jira application, the application must first be [licensed](#), and secondly, the application must have at least one default group assigned to it. Any users added to this group will be able to log in to the application. This is called **application access**. Your Jira application may have more than one group assigned to it, and a user may be a member of more than one group assigned to the application, but they will only count as one licensed user for that application. This is covered in more detail on [Managing users access to Jira applications](#).

## On this page:

- [Installing your first application and application access](#)
- [Adding additional Jira applications](#)
- [Running multiple Jira applications](#)

## Installing your first application and application access

When you [install your first application](#) and license it (you may obtain a license as part of the installation process, or directly from [my.atlassian.com](#)), Jira will create two user groups, and add you to both of them. The first group is the **jira-administrators** group, and this is the group that grants you the **Jira Administrator global permission** and grants you administrative privileges. The second group created depends on the Jira application you have installed. They are listed below:

JIRA application	User group created when the product is licensed
Jira Core	<b>jira-core-users</b>
Jira Software	<b>jira-software-users</b>
Jira Service Management	<b>jira-servicedesk-users</b>

Both of these groups are assigned to the application you installed on the **Application access** page, and the second group is also assigned as the [default group](#). This means any subsequent users you create for the application will be added automatically to this group.

## Adding additional Jira applications

You may have a requirement to add another Jira application to your instance. You can [install additional applications](#) through your **Version & licensing** page. This allows you to locate the most up-to-date version of the application and install it. Once installed, you'll still need to ensure the new application is licensed. Once licensed, Jira will create a default group for the application, but you will not be added to this group automatically. To gain full access to the application, you should add yourself to a group associated with the application.

## Running multiple Jira applications

Each Jira application comes complete with a specific set of features and functions, which tailors the experience delivered to its users. Every user in Jira will have access to an application based on their [membership of groups](#). A user may have access to all the applications, or only one application. If a user has access to an application, they will count as a licensed user for that application. For example, if a user belongs to a group for Jira Software and a group for Jira Service Management, they will count as a licensed user for both Jira Software and Jira Service Management.

When you have multiple applications installed, by default, all users will be able to view all projects (unless there are specific project permissions set up that prohibit this). This means a Jira Core user will be able to see all Jira Software and Jira Service Management projects. However, as they are not licensed for these applications, they will not be able to see any features or functions that are specific to that application. For example, a Jira Core user viewing a Jira Software project would be able to see the project and its issues, but would not be able to see any Jira Software specific features, like Agile boards, development information, or release information. These features can only be viewed by a Jira Software user. It's important to note that Jira Core does not have any specific features or functions that cannot be viewed and/or actioned by other users. This means that if you are a Jira Software or Jira Service Management user, you can already view and work on a Jira Core project. You do not need to have specific application access for Jira Core, and therefore do not need to consume a license. View the [Jira applications and project types overview](#) page for more information on what licensed users can and cannot view and action on projects from other applications.

# License compatibility

Each Jira application you install must have a unique license. There are various license types available. Some of these license types are incompatible with each other. If you try to install incompatible license types, Jira will present you with an error. To resolve this, you should select compatible license types, obtain them and install them. You should make sure you remove the incompatible license type first.

You can manage your Jira licenses on the [Versions & licenses](#) page.

## Commercial licenses

A [commercial](#) license is a paid license that allows you to run a Jira application and add users.

- All commercial licenses will work with each other if you have more than one Jira application installed.
- You can mix commercial licenses with evaluation licenses.
- You cannot mix commercial licenses with other license types (e.g. Data Center, Server, or Academic licenses).

## Data Center licenses

[Data Center](#) is the enterprise edition of Jira.

- If you have installed a Data Center license for an application and configured the application for Data Center, all subsequent licenses must be Data Center licenses.
- If you have any other type of license and want to install a Data Center license, this can be done.
- If you have more than one Jira application, and you want to set them up for Data Center, all the applications must have Data Center licenses. You cannot mix a Data Center license with any other type. For example, you can't use a Jira Service Management Server license with a Jira Software Data Center license.

### Moving from Server to Data Center

If you have Jira Service Management installed on your Jira Software instance, and you enter a Data Center license for one of the products, we prompt you to update your other license at the same time.

## Evaluation licenses

An evaluation (or "trial") license lets you try the full functionality of a Jira application for a fixed period of time (typically 30 days). When the trial ends, the application stops functioning until you install a paid license.

## Unpaid licenses

Unpaid licenses are available for evaluators, not for profit organizations, charities and students.

- All unpaid licenses will work with each other if you have more than one Jira application installed.
- You cannot mix unpaid licenses with commercial (paid) licenses when you have more than one Jira application installed.

### Move from Server to a non-clustered Data Center deployment option

Your Data Center license must match the one you used for Server. For example, if your Server license included Jira Software and Jira Service Management, but the new one has Jira Software only, you won't be able to update it.

## Data Center app licenses

Atlassian Data Center products require you to use Data Center apps, if a Data Center equivalent is offered by the app vendor.

If you're upgrading your host product (for example, Jira Software or Jira Service Management) from a Server license to a Data Center license, you may also be required to upgrade any Server apps installed on your instance. This involves upgrading your license, and potentially updating to a newer version of the app.

Learn more about [upgrading your Server apps after moving to Data Center](#)

# Extending Jira applications

Jira is very flexible, and has a number of extension points where Jira's data can be queried or its functionality extended. This page provides an overview of the mechanisms available for extending Jira.

**i Jira apps:** For information on installing or enabling existing apps, please read the [Managing apps](#) documentation. To learn about creating your own apps, see [developing apps with the Atlassian Plugin SDK](#).

Note that an app that specifically plugs into the architecture of an Atlassian application such as Jira is sometimes called a **plugin**, although the terms 'plugin' and 'app' are often used interchangeably.

Custom field types	Jira comes with various <a href="#">custom field types</a> defined. New types can be written and plugged into Jira. See the <a href="#">How to create a new Custom Field Type</a> tutorial for more information.
User formats	Jira comes with many options to change the <a href="#">look and feel</a> of features in the system. User formats are a feature that can be customized by apps. You can write your own user format app to change the display of user details in JIRA, e.g. display a profile picture. See the <a href="#">User Format Plugin Module</a> for more information.
Gadgets	New <a href="#">gadgets</a> can be created by writing an XML descriptor file, packaged as an <a href="#">Atlassian app</a> . See <a href="#">Tutorial - Writing gadgets for Jira</a> for more information.
Reports	Jira comes with various reports built-in. Using the app system, new reports can be written, providing new ways of viewing and summarizing Jira's data.
Workflow functions and conditions	Jira's issue workflow (states and state transitions an issue can go through) can be customized through the web interface (see the <a href="#">workflow documentation</a> ). The workflow engine provides hooks where you can plug in your own behavior: <ul style="list-style-type: none"> <li>• Run arbitrary Java when a certain transition occurs, via post-functions.</li> <li>• Limit visibility of transitions to certain users, via conditions.</li> <li>• Validate input on transition screens (eg. in comments), via validators.</li> </ul> <p>See the <a href="#">Working with workflows</a> for details on workflow post-functions, conditions, and validators. Once written, these can be packaged as apps and reused.</p>
Issues and projects	On the 'View Issue' page, some issue information (comments, change history) is displayed. Likewise, the 'Browse Project' page contains separate sections, listed on the far left, for different types of project information. <p>By writing an app, you can add new issue or project sections (that will be listed in the left panel) to Jira. For instance, you may wish to display project/issue data pulled in from an external source. This is how the <a href="#">Jira Subversion app</a> works.</p>
Listeners (Note this is not configurable in Jira Cloud applications)	Jira has a complete event subsystem, which fires events whenever anything happens. For example, an <code>ISSUE_CREATED</code> event is fired whenever an issue is created. A listener is just a class that implements a <code>JiraListener</code> interface, and is called whenever events occur in Jira. Using those events, you can then perform any action you want. For example the email sent by Jira is driven by the <code>MailListener</code> . This is useful when you want to drive or affect external systems from events, which occur within Jira usually used to <i>push</i> data into outside systems. For more information, read the <a href="#">listeners documentation</a> .
Services	Services are classes that implement the <code>JiraService</code> interface. When installed, you specify an update period, and Jira will call the <code>run()</code> method of your service periodically. A sample service is <code>POPCommentService</code> . This service checks a particular POP mailbox periodically, and if it finds messages, tries to extract an issue key from the subject. If the subject contains a key, the body of the mail is added as a comment to the message. Services are useful when you want to periodically <i>pull</i> data into Jira from outside systems. For more information, see the <a href="#">services guide</a> .



# Integrating with development tools

Connecting Jira Software to compatible development tools provides your team with a range of functionality and information related to your development work. You can connect to multiple instances of the same development tool, but its recommended you set up one of these instances as the primary link so Jira Software queries that instance first when looking for that sort of information.

Jump to...

- [Available features](#)
- [How it works](#)
- [Viewing integrated dev tools](#)
- [Integrating with dev tools: DVCS or app links](#)

## Available features

Features that you can use depend on your development tool. Heres a brief summary:

### Development panel on issues

The Development panel is shown on the View Issue screen, and provides the following functionality:

- **Bitbucket Cloud and Bitbucket Server:** view and create branches, view commits, and view and create pull requests
- **Fisheye / Crucible:** view commits and branches, view and create reviews
- **Bamboo:** view the status of builds and deployments
- **GitHub / GitLab:** view commits, branches, and pull requests

**Development**

2 branches	Updated 14/Jan/14 11:51 AM
4 commits	Latest 14/Jan/14 8:44 PM
1 pull request <b>MERGED</b>	Updated 14/Jan/14 8:44 PM
3 builds	Latest 21/Jan/14 11:51 AM

Deployed to Production

[Create branch](#)

For more info on using the development panel in issues, see [Viewing the development information](#).

### Workflow triggers

Workflow triggers can help keep Jira Software issues synchronized with the information in your development tools. Instead of relying on developers to manually update the status of an issue after committing code, completing reviews, or creating branches, you can configure triggers in your workflow to automatically transition issues when these events occur in your development tools. For example, you could configure a trigger to automatically transition an issue from 'To Do' to 'In Progress' when a branch is created.

## Add trigger



**Pull request created**  
Automatically transitions the issue when a related pull request is created in a...



**Pull request merged**  
Automatically transitions the issue when a related pull request is merged in a...



**Pull request declined**  
Automatically transitions the issue when a related pull request is declined in a...



**Pull request reopened**  
Automatically transitions the issue when a related pull request is reopened in a...



**Branch created**  
Automatically transitions the issue when a related branch is created in a connected...



**Commit created**  
Automatically transitions the issue when a related commit is made in a connected...



**Tell us what other triggers you'd like to see**  
We are very interested in how you'd like to automate your workflow.

Next
Cancel

For more info on workflow triggers, see [Configuring workflow triggers](#).

## Release Hub

The Release Hub shows the progress of a version, so you can determine which issues are likely to ship at a glance. The commits, builds, and deployments related to each issue are shown, helping you to spot potential development issues that could cause problems for a release.

When you are ready, you can also release the version from the Release Hub. Doing this marks the version as complete, moves incomplete issues to other versions, and triggers release builds (if Jira Software is integrated with Bamboo).

Version 2.0 UNRELEASED
Release

Start: 03/Nov/14 Release: 15/Feb/15 [Release Notes](#)

48 Warnings

273 Issues in version

262 Issues done

7 Issues in progress

4 Issues to do

Warnings indicate when the status of a JIRA issue doesn't reflect related development activity. For example: an issue marked complete that has an open pull request should be marked as still being in progress. Manage Warnings

**Unreviewed Code**  
These issues have been marked complete but the commits are not part of a pull request or review.

1-4 of 4 View in Issue Navigator

P	T	Key	Summary	Assignee	Status	Development
⊘	⊠	SSP-1663	UI is not loading in IE8	 Andrew Swan	DONE	1 commit <span style="color: green;">✔</span> <span style="color: green;">🚀</span>
⊘	⊠	SSP-1979	Incorrect permissions for new report	 Andrew Swan	DONE	1 commit <span style="color: green;">✔</span> <span style="color: green;">🚀</span>
⬇	⬆	SSP-1555	As a developer, I want to view the build status for an issue	 Bruce Templeton	DONE	5 commits <span style="color: green;">✔</span> <span style="color: green;">🚀</span>
⬇	⊠	SSP-1660	Missing error message when Bamboo is unavailable	 Eduardo Soares	DONE	1 commit <span style="color: green;">✔</span> <span style="color: green;">🚀</span>

1-4 of 4

For more info on the Release Hub, see [Checking the progress of a version](#).

## How it works

When the Atlassian development tools are integrated with Jira Software, a user simply needs to supply an issue key for the issue to be automatically linked:

- **Commits** are linked automatically if the issue key is included in the commit message.

- **Branches** are linked automatically if the issue key is included in the branch name.
- **Pull requests** are linked automatically if the issue key is included in the pull request's title or in the source branch name.
- **Reviews** are linked automatically if the issue key is included in the title of the review, or if the [issue is linked](#) from the review.
- **Builds and deployments** are linked automatically if a commit involved in the build has the issue key in its commit message.

When triggers are configured in the workflow for your project, particular events published by the developer tools automatically transition issues.

For more info on how to properly reference issues, see [Referencing issues in your development work](#).

### Viewing integrated dev tools

Development tools would usually be integrated by Jira System Administrators, but project admins can also check which tools are already integrated and available to use for their projects.

To view available development tools:

1. Open your project.
2. Go to **Project settings > Development tools**.

**Connected developer tools** Connect ▾

Development information will be displayed from these connected applications. [Learn more](#).

**⚠ Capabilities Warning**  
 Applications with flagged capabilities may be offline or older versions. Upgrade flagged applications to access more features.  
 The table below shows application compatibility with the Development panel - older applications listed as incompatible will continue to display information in the usual locations

Name	Application	Application URL	Capabilities	Refresh ↻
atlaseye	FishEye / Crucible	<a href="https://atlaseye.atlassian.com">https://atlaseye.atlassian.com</a>	Guaranteed delivery Create reviews View branches View commits View reviews	
Bitbucket Cloud	Bitbucket Cloud	<a href="https://bitbucket.org">https://bitbucket.org</a>	Create branches Create and view pull requests View commits	
Bitbucket MAD	Bitbucket Server	<a href="http://mad-mad-mad.localtunnel.me/bitbucket">http://mad-mad-mad.localtunnel.me/bitbucket</a>	Smart Commit producer Create branches Create and view pull requests View commits	

### Integrating with dev tools: DVCS or app links

Depending on what application you're using for development, you'll integrate it differently either through the DVCS account page in Jira, or through app links. You can also have both types of integration.

Application	Integration type
<ul style="list-style-type: none"> <li>• Bitbucket Cloud</li> <li>• GitHub / GitHub Enterprise</li> <li>• GitLab / GitLab self-managed</li> </ul>	<a href="#">Integrate using DVCS</a>
<ul style="list-style-type: none"> <li>• Bitbucket Server</li> <li>• Fisheye / Crucible</li> <li>• Bamboo</li> </ul>	<a href="#">Integrate using app links</a>

# Integrating with development tools using DVCS

The DVCS accounts page is one of the integration options for development tools that lets you link your Bitbucket Cloud, GitHub, and GitLab accounts to Jira. Once linked, you can view the development information from your repositories directly in your Jira issues. [Learn more about integrating with development tools](#)

## Linking your accounts

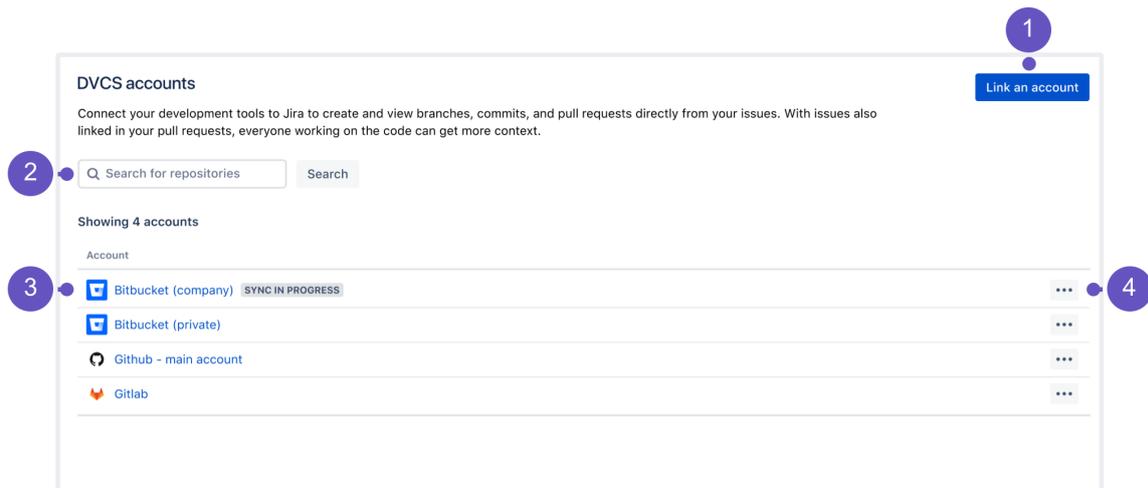
You can link your Bitbucket Cloud, GitHub, and GitLab accounts to Jira. This is similar to integrating an external application, so you'll also need to configure some items in your development tools, like OAuth access tokens.

- [Linking Bitbucket Cloud accounts](#)
- [Linking GitHub accounts](#)
- [Linking GitLab accounts](#)

**i** If you're using other development tools that aren't listed here, like Bitbucket Server or Bamboo, you can still [integrate them through app links](#).

## Managing your accounts

Once you link your accounts, they will appear on the accounts list, together with their repositories.



1. **Link an account:** Add your account, and its repositories, to Jira.
2. **Search:** Search for a repository to get a list of repos and available configuration for them.
3. **Accounts:** After linking an account, it appears on the list. Click any of the accounts to view details about its repositories.
4. **Configuration:** Configuration options for your whole account.

## Configuration options

For each account, you can complete these actions:

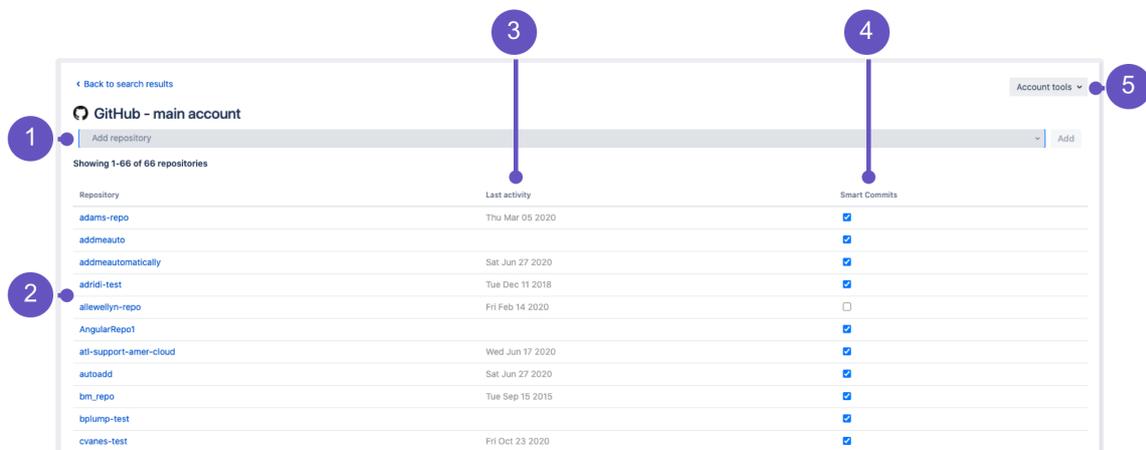
Setting	Description
Visit account page	<i>Not available for Bitbucket Cloud</i> Opens the external account page.

<b>Configure default account settings</b>	Allows you to set some global settings for your account: <ul style="list-style-type: none"> <li>• <b>Automatically sync new repositories:</b> Keep your repositories in Jira up to date with any new changes.</li> <li>• <b>Enable Smart Commits:</b> Allows you to process your issues using special commands in your commit messages. For more info, see <a href="#">Processing issues with Smart Commits</a>.</li> </ul>
<b>Refresh repositories</b>	Synchronizes changes from all repositories in this account using soft sync, which means that only changes that happened since the last synchronization are updated. You can also perform a full sync (recreating data) from the repositories view.
<b>Configure automatic access</b>	<i>Available only for Bitbucket Cloud</i> For more info, see <a href="#">Configure automatic team invitations</a> .
<b>Reset OAuth settings</b>	<i>Not available for GitLab and GitLab self-managed</i> Resets the settings that you specified when linking an account.
<b>Delete</b>	Deletes your account from Jira.

## Managing your repositories

You can view your repositories in two ways:

- **Search:** Use the search bar on the main DVCS accounts page to find a list of repositories and view actions for them.
- **Account:** Click any of your linked account to switch the view to repositories.



1. **Add repository:** Add an unlinked repository to this account. When adding a new account, you can choose to auto-link all repositories, in which case you'd already have all of them on the list.
2. **Repositories:** List of your repositories. Click any to open it in your development tool.
3. **Last activity:** Information about the last activity. When you hover over it, you'll see a refresh icon that lets you sync the latest changes. **Good to know:** You can also shift-click this icon to view additional syncing options (full sync).
4. **Smart Commits:** Enables or disables Smart Commits for this repository.
5. **Account tools:** Configuration options for your whole account.

## Viewing your development information in Jira

Once you've linked your accounts and their repositories, Jira will automatically search them for issue keys to try to connect your development work with relevant issues.

For more info on how to reference issues in the development work, and how to view the dev info in issues, see:

- [Referencing issues in your development work](#)
- [Viewing the development information for an issue](#)

# Linking Bitbucket Cloud accounts

You can link your Bitbucket Cloud account and its repositories to Jira, and see the development information from pull requests displayed in your issues.

**Prerequisite:** Create the OAuth access token in Bitbucket Cloud

First you need to create the OAuth access token in Bitbucket Cloud to give Jira permissions to access it.

 You need to be logged in as a user with administrative rights to the account in Bitbucket Cloud.

To create the OAuth access token in Bitbucket Cloud:

1. Select **avatar** > **All workspaces**.
2. Select the **workspace** the connector is going to be added to.
3. Select **Settings** > **Apps and features** > **OAuth consumers**.
4. Select **OAuth** under **Access management**.
5. Select **Add consumer**.
6. Enter the following details:
  - **Name:** A descriptive name, for example Jira DVCS.
  - **Description:** A helpful reminder of the purpose of this token.
  - **URL:** A URL for your Jira Software instance, for example <https://example.atlassian.net>
7. Select the following permissions:
  - **Account:** Write
  - **Repositories:** Admin (but not Repository: Write)
  - **Pull requests:** Read
  - Additionally, select the **This is a private consumer** checkbox.
8. Select **Save**.
9. Select the name of your new consumer to see the **OAuth key** and **OAuth Secret**. You'll need them when adding your Bitbucket Cloud account in Jira.

## Link your Bitbucket Cloud account

To add a Bitbucket Cloud account in Jira:

1. Go to **Administration** > **DVCS accounts**.
2. In Host, select **Bitbucket Cloud**.
3. Enter the **Team or User account** name.

### Example

If you want to link the account that owns the <https://bitbucket.org/tutorials/markdowndemo> repository, you would enter `tutorials` as your account. Linking the `tutorials` account links all of that account's repositories, not only the `markdowndemo` repository.

4. Copy **OAuth Key** and **OAuth Secret** from Bitbucket Cloud, and enter them as **Client ID** and **Client Secret**.
5. Leave the default auto link and smart commits (recommended) as is or change them.

6. Select **Add**.

### Add New Account

Host

OAuth Key\*   
[Help with my key and secret.](#)

OAuth Secret\*

Auto Link New Repositories

Enable Smart Commits  
[Transition Jira issues](#) through commit messages.

### Whats next?

Heres what happens after adding your account:

- **Syncing repositories:** Your account appears on the DVCS page and Jira starts syncing its repositories, if you decided to link them automatically. For details, see [Integrating dev tools using DVCS](#).
- **Matching issues:** If your commits and pull requests include issue keys, Jira will try to find them and add the relevant information to your issues. For details, see [Referencing issues in your development work](#).
- **Displaying dev info:** If referenced correctly, the dev information will be added to your issues to give everyone involved more context. For details, see [Viewing the dev info for an issue](#).
- **Workflow triggers:** One of the benefits of integrating your development tools is adding workflow triggers that, for example, lets you change the issue status after you create a new branch. For details, see [Configuring workflow triggers](#).

# Getting started with Bitbucket and Jira Cloud

Learn how you can connect the Bitbucket code hosting service with Jira Cloud. Connecting Bitbucket to Jira gives your team the power to see commits, branches, pull requests. You can also create branches and see the status of pull requests all from the development panel in Jira or Jira Agile.

**Development**

---

1 branch

14 commits Latest 22/Aug/16 9:47 AM

1 pull request OPEN Updated 2 hours ago

1 build ✔ Latest 22/Aug/16 10:00 AM

---

[Create branch](#)

You won't see the builds reference shown in the screenshot above unless you [Integrate Jira with Bamboo](#).

## On this page:

- [Before you begin...](#)
- [Step 1. Sign up for Bitbucket and create a Bitbucket team](#)
- [Create a team](#)
- [Step 2. Invite team members to your team on Bitbucket](#)
- [Step 3. Create repositories in or move repositories to your Bitbucket account](#)
- [Step 4. Connect the team account in Jira](#)
- [What to do next](#)

## Before you begin...

1. Bitbucket and Jira Cloud are independent services: you will have both a Jira Cloud account and a Bitbucket team each with their own set of users, permissions, and access rules.
2. Bitbucket teams are not accounts: they must be managed by an administrator (or administrators) who have individual Bitbucket accounts. However, the Bitbucket team can have its own payment plan.
3. Every member of the Bitbucket team must have their own individual Bitbucket account. When you invite new team members using their email they are also automatically invited to sign up for Bitbucket and are automatically added to your team when they complete sign up.
4. You can transfer Bitbucket [repository ownership to a team](#): this can be helpful if you want to create a team based upon existing repositories.

## Step 1. Sign up for Bitbucket and create a Bitbucket team

If you don't already have Jira Cloud, [set up](#) a trial or a paid instance.

### Create a Bitbucket account

If you already have an account, you can skip this section and go to the [next](#). When you create a Bitbucket individual account you must supply the following fields:

Field	About what you are supplying
-------	------------------------------

<b>User name</b>	Up to 30 character username. You can use letters, numbers, and underscores in your username. Your username must be unique across the entire Bitbucket site.  Bitbucket appends this username to the URL for all the repositories you create. For example, the username <code>atlassian_tutorial</code> has a corresponding Bitbucket URL of <a href="https://bitbucket.org/atlassian_tutorial">https://bitbucket.org/atlassian_tutorial</a> .
<b>Email address</b>	An email address that is unique across the entire Bitbucket site. The system sends you a confirmation email.
<b>Password</b>	A combination of up to 128 characters. If you are using a Google account to sign up the system uses that password. You are responsible for ensuring that your account password is sufficiently complex to meet your personal security standards.

To sign up for a Bitbucket account:

1. Open <https://bitbucket.org/account/signup/> in your browser.
2. Complete the fields in the sign up form.
3. Click **Sign up**.

When you are done signing in, Bitbucket places you in the **Dashboard** of your account. Take a second to look around the user interface. Across the side of each Bitbucket page is a series of options that let you navigate around Bitbucket. On the top bar is a link for **Teams**. Select **Teams > Create team** and move to the next section.

### Create a team

To better understand how teams work lets first take a look at how they fit into the Bitbucket environment.

<b>Teams are comprised of</b>	<b>Which are</b>
Users	Who develops the code and manages the team.  Each user has an individual Bitbucket account which can be added to (or removed from) any group or team within the Bitbucket universe.
Groups	What users can do and where they can go.  Groups provide permissions (administrator, read/write, read only) to groups of individual users, and are assigned to repositories for the team.
Repositories	Where the code lives.  The repository is where you store, access, create, develop, modify, and share the code for a project

### Create a team

The following process and infographic describe how to create a very simple team consisting of you, one member, and one repository. Feel free to follow along in your Bitbucket account, or just read through to get an idea of what goes into creating a team.

#### Prerequisite:

**You have an individual Bitbucket account**

**Create a team**

Two user groups, **Administrator** and **Developer**, are created by default when you create a team.

To create a team:

1. Select **Teams > Create team**.
2. Fill in the available fields:
  - a. Team name (when adding a team to a Jira Cloud instance, consider using the same name as the Jira Cloud instance or one you can easily identify with a specific project).
  - b. Team ID
3. Add additional team members by entering their Bitbucket username or email address and clicking **Add** for each person you want to invite to the new team.  
**Note:** you are already a member of the team and the administrator by default.
4. Click **Create**.

Congratulations you have a team! You are taken to the team overview page where you can create your first team repository or manage the team's settings.

**Step 2. Invite team members to your team on Bitbucket**

You can add team members to your linked Bitbucket team account. These may be the same users you added to Jira. It is your choice. Users that you add to Bitbucket need not have accounts on the Jira instance.

To add a user to a team, you add the user to one of the team's groups, as follows:

1. Log in to Bitbucket as a user with administrative rights on the team.
2. Choose Your avatar > **View all teams**.
3. Choose the team you want to add new members to.
4. Choose **Settings > User groups**.
5. Click on the group you want to add the user to.
6. Enter the user's username or email address, then click **Add**:

**Members**   **Repositories**

---

Add

	<a href="#">matthewhunter</a>	Matthew Hunter	<span style="border: 1px solid #ccc; border-radius: 50%; padding: 2px 5px;">×</span>
	<a href="#">tw-bot</a>	tw-bot	<span style="border: 1px solid #ccc; border-radius: 50%; padding: 2px 5px;">×</span>

If you entered an email address that corresponds to a Bitbucket account, Bitbucket resolves the account for you. If Bitbucket can't resolve the address, it sends the user an invitation to create a Bitbucket account and join the team.

**Step 3. Create repositories in or move repositories to your Bitbucket account**

A repository (sometimes called a repo) contains your project code. Whether you have no files or many files, you'll first want to create a repository on Bitbucket Cloud. From there, you can clone your repository to your local system and start working on it.

If you name a repository with upper case letters, you'll see the name with upper case letters in Bitbucket, but Bitbucket converts the name to all lower case in the repository UR. As a result, you can't create two repositories with names that result in the same URL.

**To create a repository**

1. Click **+** in the global sidebar and select **Repository** under **Create**.
2. Select the **Workspace** where you want to create the repository.
3. Select a project from the **Project** dropdown menu. If a project does not exist, click on **Create new project** at the bottom of the **Project** dropdown menu to create a new project in which to work and collaborate with others on your repository.
4. Enter a **Repository name** that will describe your repository and appear in its URL.
5. Keep access to your repository set to **Private** unless you want to make your repository public so that anyone can see it.
6. If you already have files that you want to add to your repository, select **No** from **Include a README?** Otherwise, go with the default option or select a one of the included README options.
7. Select the **Version control system**. If you don't know the difference, keep **Git** as the default system.
8. Click **Create**.

**After you create a repository**

What comes next depends on what you want to do with your repository:

- **Starting from scratch with no files?** Clone the repository to your local system to connect Bitbucket repository to a local directory. [Learn how](#)
- **Working on existing files that aren't under version control?** Add unversioned files to a repository before pushing them to Bitbucket. [Learn how](#)
- **Already have local files in a Git or Mercurial repository?** Push versioned code to an empty repository, maintaining commit history. [Learn how](#)

Take a minute to explore what comes with your new repository.

The Bitbucket service allows you to create an unlimited number of public repositories. The number of private repositories is restricted by your plan.

**Tips, Tricks, and Links to More Information**

- You can [transfer a Bitbucket repository](#) from an individual Bitbucket account to your Jira team account.
- You can [import a Git or Mercurial project from your local system](#) into Bitbucket.
- To learn about Bitbucket's few restrictions on repositories, see [this page](#).
- Some users have security and backup concerns about code, see [this page](#) for details.
- See the Atlassian blog for information about [Centralized vs. Distribute Version Control System \(DVCS\)](#).

**Step 4. Connect the team account in Jira****Make sure you understand how Jira connects to your Bitbucket account**

The connector needs permission from your Bitbucket account to access your account data. The connector does this through an [OAuth](#) access token.

You create an OAuth access token from the Bitbucket account. You should create the access token on the team that owns the repositories that you want to link. The values that make up the token are:

key	A string generated by the Bitbucket system.
secret	A string generated by the Bitbucket system.
authorizing account	The account that authorizes the token.

After you create a key and secret in Bitbucket, go back to Jira. There, you can enter the account, the OAuth key, and secret data.

Bitbucket does not automatically trust the key and secret it will ask you to authorize the Bitbucket connection.

When you link your Bitbucket account with Jira all the public and private repositories owned by the account. It adds a POST commit hook service to the repository on the Bitbucket system. The POST commit hook is a piece of code that sits on the repository waiting for users to commit changes.

On the Jira Cloud side, the repositories owned by your Bitbucket account appear on the **Manage DVCS Accounts** page. A team member may create repositories under their individual Bitbucket account, but assign the team as the owner. These repositories also appear in Bitbucket under the list.

### Procedure to link an account

It is a two step procedure to link a Bitbucket account to Jira. To work through this procedure, you must have administrative rights on both the Jira Cloud instance and on the Bitbucket account you want to link.

#### Step 1. Create an OAuth access token for your Bitbucket account

To link a Bitbucket account you create an OAuth access token on your Bitbucket account. If you are linking repositories under a team, you should generate this token under the team account.

Log in to Bitbucket as a user with administrative rights on the account.

1. Choose **Manage account**.
2. (Optional) If connecting a team, choose the team from the **Account** drop-down.
3. Select **OAuth**.
4. Click **Add consumer**.
5. Enter `Jira DVCS` for the **Name**.
6. Leave the other fields blank.
7. Press **Add consumer**.

Keep your browser open to Bitbucket and go onto the next step.

#### Step 2. Link the account on Jira

To complete the link between your DVCS and Jira:

1. Log in to Jira Cloud as a user with administrative rights.
2. From the Jira dashboard click the settings icon (⚙️).
3. Choose **Applications** then **DVCS accounts** (under 'Integrations' in the left-hand panel).
4. Click **Link a Bitbucket Cloud or GitHub account**.
5. Choose **Bitbucket Cloud** as your **Host** value.
6. Enter a **Team or User Account**.

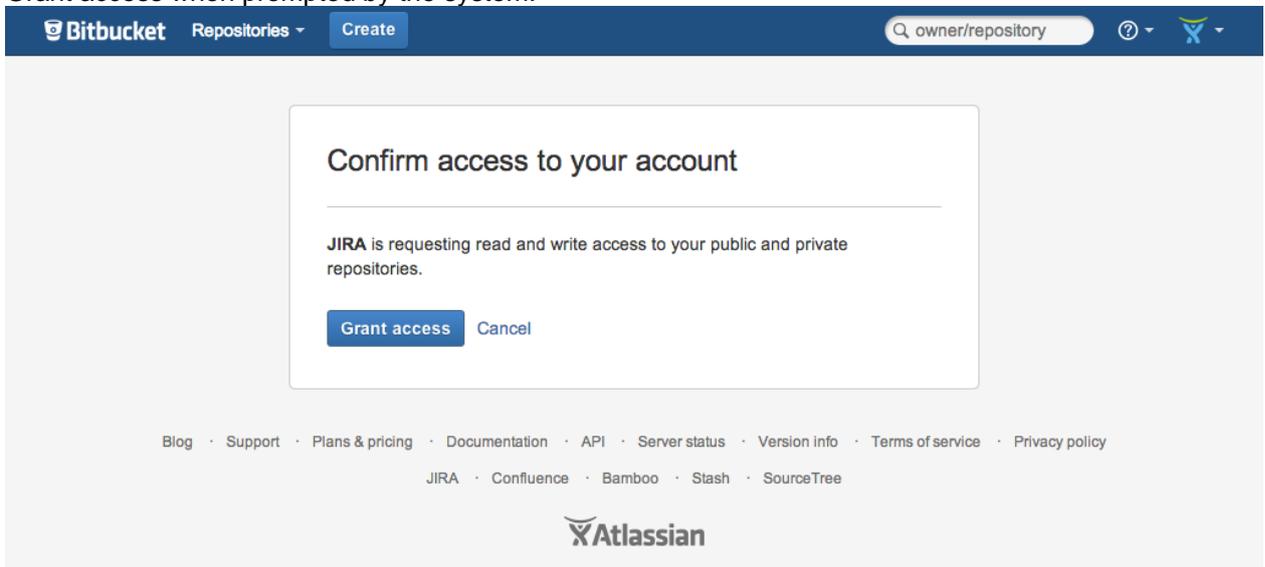
For example, if you want to link the account that owns the <https://bitbucket.org/tutorials/markdowndemo> repository then you would enter `tutorials` for the **Team or User Account** value. Linking the `tutorials` account links all of that account's repositories, not only the `markdowndemo` repository.

7. Copy the **OAuth Key** and **OAuth Secret** from your Bitbucket account into the dialog.

8. Leave the default auto link and smart commits (recommended) as is or change them.

9. Click **Add**.

10. Grant access when prompted by the system:



11. Upon success, the **DVCSaccounts** page displays with your account.

The account you just connected and all of its repositories appear in the **DVCS accounts** page. The initial synchronization starts automatically. After that, the system continues to sync your repository automatically on a regular basis.

### Automatic synchronization and temporarily disabling a link

After you link an account, Jira automatically starts looking for commits that reference existing issue keys. The summary shows the synchronization results and errors, if any. A synchronization of commit data from the DVCS repository to Jira can take some time. As the synchronization progresses, the commits appear in related issues. You can always enable and disable the linking of repositories with Jira as needed.

### How the link appears in Bitbucket

The DVCS Connector does two things:

- It adds an OAuth consumer to the linked account's list of integrated applications. To view the listing in Bitbucket, click your profile image and select **Manage Account**. Click **Integrated applications** and you'll see a listing similar to the following:

- The DVCS Connector programmatically adds a POST commit hook service to each of the account's repositories. To view this service, choose **Settings** (⚙️), then click **Hooks** to display the Hooks page. You'll see a listing similar to the following:

The DVCS Connector uses its link to check for new repositories on the account, then adds this service to those as well. You see the result of all this on the **Services** page.

## What to do next

What you do after getting started depends on your team's own knowledge and needs:

- If your team is unfamiliar with code hosting using Bitbucket or brand new to DVCS, [you should work through the Bitbucket 101](#).
- If your team is comfortable with DVCS and Bitbucket, you might want to learn to [Link to a web service](#) in Bitbucket which will give you even more interaction between Jira and Bitbucket.
- The DVCS Connector lets you [update and move Jira issues through a workflow](#), see [processing Jira issues with smart commit messages](#) for commands and examples of how to do this.

# Configure automatic team invitations

You can add users to a Bitbucket group through either Bitbucket or Jira Software **Server**. These groups are defined in a Bitbucket account (workspace). When you add users, they receive an email invitation to join Bitbucket and the account group. To join a group, users must already have their own individual Bitbucket accounts. Each invitation contains a link that takes a user to Bitbucket, where they finish joining by providing their account (or by creating an account if necessary).

When you add users through Bitbucket, you can supply a user's Bitbucket account name or their email. If you are using Jira Software Server, you have options for adding Jira Software Server users manually or automatically (by a public signup). These users receive invitations only from the Bitbucket accounts you've configured to send them.

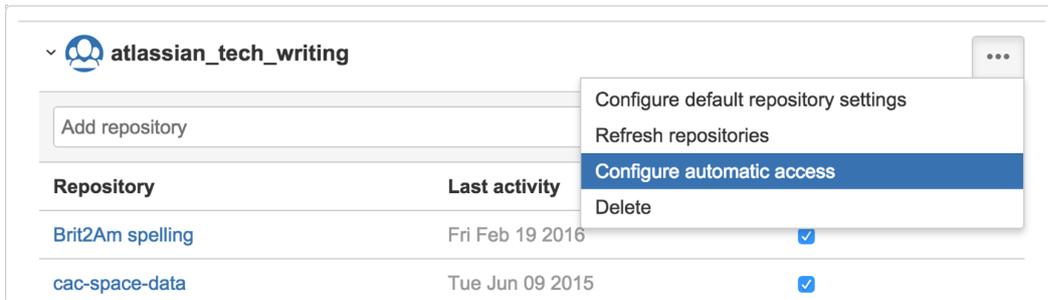
## Decide on a configuration strategy for adding users to account groups

Before you configure automatic user sign up, decide which strategy for adding users to Bitbucket Cloudworks best for your situation.

Which description best matches your situation?	The configuration we recommend you should use:
All of my Jira Software users should belong to one or all of my Bitbucket Cloud groups.	Automatically invite new Jira Software users to a Bitbucket Cloud account's group when they become Jira Software users.  See the <a href="#">following section</a> on this page.
I add my users manually to my Jira Software Server instance.	Automatically invite new Jira Software users to a Bitbucket account's group. Optionally, change the Bitbucket groups for the user when you add them to Jira Software.  See the <a href="#">following section</a> on this page.
I've enabled automatic sign up on Jira Software but only a small set of Jira Software users should be a member of this workspace.	Prevent the automatic invitation of new users. Manually add users to groups through Bitbucket.  See the <a href="#">following section</a> on this page.
I've enabled automatic sign up on Jira Software but some Jira Software users should belong to one Bitbucket group in a specific workspace and other users need to belong to another.	

## Set the groups that users are automatically invited to join

1. Log into Jira as a user with administrative rights.
2. Go to **Applications > DVCS accounts** in the Jira admin area.
3. Locate the account to configure. If you don't see your account, click **Link Bitbucket Cloud account** to start the connection process.
4. Choose **Configure automatic access** from the Actions menu:



5. Select the groups you want new Jira users assigned to, in the 'Configure automatic access' dialog. New Jira Software Server users will automatically be added to the groups you select, and will be invited to join Bitbucket. When the user joins, they have the group access to your project. To prevent users from being invited to join groups, deselect those groups.
6. Press **Save**.

# Linking GitHub accounts

You can link your GitHub and GitHub Enterprise accounts and their repositories to Jira, and see the development information from pull requests displayed in your issues.

## Supported versions

Here's a list of supported versions:

Jira version	GitHub	GitHub Enterprise
8.14+	Current	13.0+
7.0+	Current	11.10.290+

Prerequisite: Create the OAuth access token in GitHub

First you need to create the OAuth access token in GitHub to give Jira permissions to access it.

 You need to be logged in as a user with administrative rights to the account in GitHub.

1. Choose **Edit Your Profile**.
2. Select **Applications**.
3. Choose **Register new application**.
4. Enter `Jira DVCS` for the **Application Name**.
5. Enter the Jira Software URL for both the **URL** and **Callback URL** fields. Press **Register Application**.

 Make sure you enter the **Jira Software Base URL** (for example, <https://example.atlassian.net>) for both the **Homepage URL** and **Authorization callback URL** fields. *Don't* use the dashboard URL (<https://example.atlassian.net/secure/Dashboard.jspa>).

6. Keep your browser open to your DVCS and go to the next step.

## Link your GitHub account

To add a GitHub account in Jira:

1. Go to **Administration > DVCS accounts**.
2. In **Host**, select **GitHub** or **GitHub Enterprise**, depending on what you're using.
3. Enter the **Team or User account** name.
4. Copy the GitHub's **OAuth Key** and **OAuth Secret**, and enter them as **Client ID** and **Client Secret**.
5. Leave the default auto link and smart commits (recommended) as is or change them.
6. Click **Add**.

 If you get redirected to a blank page, see [DVCS connection to GitHub produces blank page](#).

### Add New Account

Host

Team or User Account\*

Client ID\*   
[Help with my key and secret.](#)

Client Secret\*

Auto Link New Repositories

Enable Smart Commits  
[Transition Jira issues](#) through commit messages.

## Whats next?

Heres what happens after adding your account:

- **Syncing repositories:**Your account appears on the DVCS page and Jira starts syncing its repositories, if you decided to link them automatically. For details, see [Integrating dev tools using DVCS](#).
- **Matching issues:**If your commits and pull requests include issue keys, Jira will try to find them and add the relevant information to your issues. For details, see [Referencing issues in your development work](#).
- **Displaying dev info:**If referenced correctly, the dev information will be added to your issues to give everyone involved more context. For details, see [Viewing the dev info for an issue](#).
- **Workflow triggers:**One of the benefits of integrating your development tools is adding workflow triggers that, for example, lets you change the issue status after you create a new branch. For details, see [Configuring workflow triggers](#).

# Linking GitLab accounts

You can link your GitLab and GitLab Enterprise accounts and their repositories to Jira, and see the development information from pull requests displayed in your issues.

## Supported versions

Here's a list of supported versions:

Jira version	GitLab	GitLab self-managed
8.14+	Current	13.0+

## Prerequisite: Integrate Jira with GitLab using OAuth 2.0

Before you can add your GitLab account to the DVCS accounts page, you need to integrate Jira with GitLab using OAuth 2.0.

 To create the OAuth 2.0 integration in Jira, you need to be a Jira System Administrator.

## In Jira

To add a new OAuth 2.0 integration in Jira:

1. Go to **Administration > System > OAuth 2.0**.
2. Select **Add new integration**.
3. In Service provider, select **Custom**.
4. Enter your integrations name.
5. Don't worry about the **Client ID** and **Client Secret** right now. You'll get them from GitLab in the next steps.
6. Enter the authorization and token endpoint, as shown in the table below.

GitLab type	Authorization endpoint	Token endpoint
<b>GitLab.com</b>	https://gitlab.com/oauth/authorize	https://gitlab.com/oauth/token
<b>GitLab self-managed</b>	<GITLAB_SELF_MANAGED_URL>/oauth/authorize	<GITLAB_SELF_MANAGED_URL>/oauth/token

- Copy the **Redirect URL**. You'll use it in GitLab in the next step.

### Add new OAuth 2.0 integration

Select the service provider for your OAuth 2.0 integration.

Service provider

Custom ▼

**Application data**

This is used to identify the integration in your application.

Name \*

Gitlab

Enter a unique name for this integration.

Description

Enter a meaningful description for this integration.

**Provider data**

You receive this data from your service provider. For more on the data you require, check the service provider's developer documentation.

Client ID \*

6f047a20378f3053867a63d381a1a0b122dbbd2c9d43494e5091ba16304bc19b

Enter the client ID created for your application.

Client secret \*

.....

Enter the client secret created for your application.

Scopes \*

a... x ⊗ ▼

Enter scopes to define the application access level to the service provider.

Authorization endpoint \*

https://gitlab.com/oauth/authorize

Enter the HTTPS URL for authorizing the integration. Pre-filled for Google and Microsoft.

Token Endpoint \*

https://gitlab.com/oauth/token

Enter the HTTPS URL for receiving access tokens. Pre-filled for Google and Microsoft.

Redirect URL

http://localhost:2990/jira/plugins/servlet/oauth2/client/callback/sZOLZjF4qGFm6lgJaVMP7A Generate Copy

Register this redirect URL on the service provider's site to complete the integration.

Cancel Save

## In GitLab

Leave your Jira configuration open and move to GitLab to create a new application. This will give you Application ID and Application Secret that you'll use in Jira.

- In GitLab, go to **User Settings > Applications**, and add a new application. Use the following data:
  - Redirect URI:** Enter the URL you copied from Jira.
  - Confidential:** Keep that enabled.
  - Scopes:** Select api. This is required for the DVCS connection to work.

User Settings > Applications

### Applications

Manage applications that can use GitLab as an OAuth provider, and applications that you've authorized to use your account.

**Add new application**

**Name**  
Jira

**Redirect URI**  
https://www.dummyurltobereplacewithrealurl.com

Use one line per URI

**Confidential**  
The application will be used where the client secret can be kept confidential. Native mobile apps and Single Page Apps are considered non-confidential.

**Scopes**

**api**  
Grants complete read/write access to the API, including all groups and projects, the container registry, and the package registry.

2. After creating the application, you should get the Application ID and Application Secret, like on the image below.

User Settings > Applications > Jira

The application was created successfully. ×

**Application: Jira**

Application ID	6f047a20378f3053867e <span>📄</span>
Secret	21e20cfaaa2ebfde9f90 <span>📄</span>
Callback URL	https://www.dummyurltobereplacewithrealurl.com
Confidential	Yes
Scopes	• api (Access the authenticated user's API)

[Edit](#) [Destroy](#)

### In Jira again

Go back to your configuration in Jira, and enter the missing information to complete the integration.

1. Enter GitLabs Application ID as Client ID.
2. Enter GitLabs Application Secret as Client Secret.
3. In Scopes, select api.

4. Save the integration, and then select **Test Connection** to make sure everything is right.

### Add new OAuth 2.0 integration

Select the service provider for your OAuth 2.0 integration.

Service provider

Custom ▼

#### Application data

This is used to identify the integration in your application.

Name \*

Gitlab

Enter a unique name for this integration.

Description

Enter a meaningful description for this integration.

#### Provider data

You receive this data from your service provider. For more on the data you require, check the service provider's developer documentation.

Client ID \*

6f047a20378f3053867a63d381a1a0b122dbbd2c9d43494e5091ba16304bc19b

Enter the client ID created for your application.

Client secret \*

.....

Enter the client secret created for your application.

Scopes \*

a... x ✕ ▼

Enter scopes to define the application access level to the service provider.

Authorization endpoint \*

https://gitlab.com/oauth/authorize

## Link your GitLab account

With the OAuth 2.0 integration created, you can use it when linking your GitLab account on the DVCS accounts page.

1. In Jira, go to **Administration > Applications > DVCS Accounts**.
2. Click **Link an account**.
3. In the **Host** drop-down, select **GitLab** or **GitLab Self-Managed**.
  - For GitLab Self-Managed, enter your GitLab **Host URL**.
  - For Gitlab, the **Host URL** is fixed to <https://gitlab.com>.
4. In the **Team or User Account**, enter the target group and optional subgroups.
  - Groups must be entered as groupA/sub-groupA/sub-sub-groupA.

 You can only have one group active as a configuration at one time. For example:

    - If an existing integration is already using groupA, you won't be able to use groupA/sub-groupA in your new integration.
    - Likewise, if an existing integration is using groupA/sub-groupA, you won't be able to use groupA in your new integration.
  - All spaces must be changed to dashes (-). For example, a group **My projects** needs to be **my-projects**. This also applies to subgroups: **My Projects > More Projects** would be **my-projects/more-projects**. Another way to get this right is to access your group on Gitlab, and copy the URL, for example <http://gitlab.com/my-projects/more-projects>.
5. In the **Client Configuration**, select the created OAuth 2.0 integration. If you have only one, it will be automatically selected.

6. Click **Add**.

### Add New Account

Host GitLab ▼

Team or User Account\*

Client Configuration Gitlab SH ▼

[Help with OAuth 2.0 client config.](#)

Auto Link New Repositories

Enable Smart Commits

[Transition Jira issues](#) through commit messages.

Add Cancel

### Whats next?

Heres what happens after adding your account:

- **Syncing repositories:**Your account appears on the DVCS page and Jira starts syncing its repositories, if you decided to link them automatically. For details, see [Integrating dev tools using DVCS](#).
- **Matching issues:**If your commits and pull requests include issue keys, Jira will try to find them and add the relevant information to your issues. For details, see [Referencing issues in your development work](#).
- **Displaying dev info:**If referenced correctly, the dev information will be added to your issues to give everyone involved more context. For details, see [Viewing the dev info for an issue](#).
- **Workflow triggers:**One of the benefits of integrating your development tools is adding workflow triggers that, for example, lets you change the issue status after you create a new branch. For details, see [Configuring workflow triggers](#).

# Integrating with development tools using app links

Using app links is one of the integration options for development tools that lets you link your Bitbucket Server, Bamboo, and Fisheye / Crucible instances to Jira. Once linked, you can view the development information in your Jira issues. [Learn more about integrating with development tools](#)

## Supported applications

Here's a list of dev tools that you can integrate, and their supported versions:

Bitbucket Server	Fisheye / Crucible	Bamboo
Bitbucket Server 4.0+ (Stash 2.10)	3.3+/3.3+	5.4+

**i** If you're using other development tools that aren't listed here, like Bitbucket Cloud or GitHub, you can [integrate them using DVCS](#).

## Integrating your development tools

To integrate your development tools:

1. In Jira, go to **Administration > Applications > Application links**.
2. Enter the application URL you want to link.

For more info on using app links, see [Using app links to link to other applications](#).

**i** If your developer tools instances are running on the same machine as Jira Software, you'll need to ensure that the applications use distinct web contexts. This avoids authentication and session issues with OAuth and application links. For example, if you were running Fisheye and Jira, you would change the default paths to:

- <https://localhost:8060/Fisheye> (rather than <http://localhost:8060/>)

Instructions:

- [Moving Bitbucket Server to a different context path](#)
- [Changing Bamboo's root context path](#)
- [Linking Fisheye to Jira](#)

Here are a few of our development tools integrated through app links:

### Configure Application Links <sup>Ⓢ</sup>

[Give Feedback](#)

Enter the URL of the application you want to link

Application	Version	Status	Actions
 <b>server-bamboo</b> PRIMARY	Bamboo 7.3.0-m103	CONNECTED	 ...
 <b>ecosystem-bamboo</b>	Bamboo 7.3.0-m103	CONNECTED	 ...
 <b>Bitbucket Server</b> PRIMARY	Bitbucket Server 7.11.0-m4	CONNECTED	 ...
 <b>server-syd-bamboo</b>	Bamboo 7.3.0-m103	CONNECTED	 ...

## Whats next?

Here's what happens after adding your account:

- **Syncing information:** Your development tools appear on the Application links page and Jira starts syncing them.
- **Matching issues:** If your commits and pull requests include issue keys, Jira will try to find them and add the relevant information to your issues. For details, see [Referencing issues in your development work](#).
- **Displaying dev info:** If referenced correctly, the dev information will be added to your issues to give everyone involved more context. For details, see [Viewing the dev info for an issue](#).
- **Workflow triggers:** One of the benefits of integrating your development tools is adding workflow triggers that, for example, lets you change the issue status after you create a new branch. For details, see [Configuring workflow triggers](#).

# Integrating with other tools

## Integrating with Flowdock

You can integrate [Flowdock](#) with Jira Cloud and issues from your Jira projects will be included in your Flowdock flows.

If you link a Jira project to a Flowdock flow, **all Jira comments will appear on FlowDock** regardless of the restriction level that is set when creating the comment. Please ensure that you only link Jira projects to Flowdock flows when it is acceptable for all Jira comments to be visible.

### To enable Flowdock in Jira:

1. Log in as an admin to your site.
2. Choose **Manage Plugins > Show System Plugins**.
3. Locate **Flowdock for Jira** and click **Configure**. The Flowdock integration page will display all the Jira projects that are set up.
4. Enter your Flowdock API key against the Jira projects that you want to include in your Flowdock flow. To get your Flowdock API key, log in to Flowdock and view the [Integrating with variety of issue trackers](#) page. Your API key will be displayed in the Jira instructions.
5. Click **Save**. The API key information will be saved and the Flowdock integration page will refresh.

You will now receive messages in your Flowdock flow for any issue activity (e.g. issue creation, issue comments added, issue fields updated, etc) in the configured Jira projects.

## Integrating with Zephyr

Jira Cloud comes with the Zephyr Enterprise Connector plugin, and this plugin sends defect metrics to Zephyr Enterprise and Zephyr Community Editions. This plugin is a different plugin from [Zephyr for Jira](#).

### To enable Zephyr in Jira:

1. Log in as an admin to your site.
2. Choose **Plugins**. You will see the list of user-installed plugins.
3. Near the bottom of the page, locate the **Zephyr Enterprise Connector** and click it to display the available options.
4. Click **Enable**. The Zephyr plugin will be enabled.

### To connect to Jira from Zephyr:

See Zephyr's [Jira Overview & Setup](#) documentation.

### Related topics

- [http://www.getzephyr.com/zephyr/test\\_management\\_integrated\\_with\\_atlassian\\_ondemand.php](http://www.getzephyr.com/zephyr/test_management_integrated_with_atlassian_ondemand.php)
- <https://plugins.atlassian.com/plugin/details/18715>

## Integrating with Subversion

Jira's Subversion integration lets you see Subversion commit information relevant to each issue. Subversion integration can be implemented by using [Atlassian FishEye](#). The FishEye integration offers [greater scalability, insight and flexibility](#) into your source code and related integration with Jira, however both solutions allow you to link Jira to related code changes in Subversion.

Revision	Date	User	Message
<a href="#">#11</a>	Fri Nov 12 17:30:39 EST 2004	Mike	Big, very exciting commit for <a href="#">TEST-3!</a> <b>Files Changed</b> <b>ADD</b> <a href="#">trunk/moved.txt</a> (from <a href="#">/trunk/copieddocument.txt #10</a> ) <b>DEL</b> <a href="#">trunk/copieddocument.txt</a> <b>ADD</b> <a href="#">trunk/NewFile.java</a> <b>DEL</b> <a href="#">trunk/mydocument.txt</a>
Revision	Date	User	Message
<a href="#">#10</a>	Thu Nov 11 18:12:59 EST 2004	Mike	Fixed <a href="#">TEST-3</a> <b>Files Changed</b> <b>MODIFY</b> <a href="#">trunk/mydocument.txt</a>

Commits will appear in this tab if the commit log mentions the issue key ('TEST-3' above).

# Listeners

Listeners are unique to Jira, and a very powerful way to extend it.

Jira has a complete event subsystem that fires events whenever anything happens inside the application. For example, an `ISSUE_CREATED` event is fired whenever an issue is created.

A Listener is a class that implements one of the Listener interfaces. It is then called whenever events occur in Jira. Using those events, you can then perform any action you want. For example, the email sent by Jira is driven by the [MailListener](#).

Listeners are most useful when you want to drive or affect external systems from events which occur within Jira.



For all of the following procedures, you must be logged in as a user with the **Jira System Administrators** [global permission](#).

## Listener interfaces

Jira has the following concrete Listeners (which extend the base `JiraListener` interface):

<code>com.atlassian.jira.event.JiraListener</code>	The base interface which all other Jira listener interfaces extend. Covers core listener properties like uniqueness, description, parameters etc. <a href="#">API doc</a>
<code>com.atlassian.jira.event.issue.IssueEventListener</code>	The main listener interface in Jira, used whenever anything happens to an issue. <a href="#">API doc</a>
<code>com.atlassian.jira.event.user.UserEventListener</code>	This listener is called whenever anything happens to a user within Jira. <a href="#">API doc</a>

## Example listeners

The examples provided may be freely used and modified for use in your own environment. The source of all examples is available and should give you good overview of how simple it is to write your own listeners. Both example listeners are included with Jira 2.1, and both implement `UserEventListener` and `IssueEventListener`.

- **DebugListener** This is a very simple listener that prints events and their content to `System.out` whenever they are received. To test this listener, add a listener with the class `com.atlassian.jira.event.listeners.DebugListener`.
- **MailListener** This listener is how mail notifications are currently sent from within Jira, and a good example of a more complex listener. It basically listens for events, and turns them into email notifications using Velocity templates to generate the mail bodies. This listener is usually always turned on in Jira - see [Email notifications](#) for more details. If you want to write more complex or more specific notifications, you can disable the internal `MailListener` and add your own.

Other examples of useful tasks that can be accomplished with listeners are:

- **Send SMS or IM notifications** A listener could easily send notifications for various events via SMS or instant messenger (e.g. ICQ or AIM) - or anywhere that you have a Java library to send messages.
- **Group notifications** A listener could notify certain groups of issue changes, depending on the content of the issue. For example any issue containing "windows" in the environment could notify your "windows-developers" group.

## Registering a listener

For custom-written listener classes, make sure your listener class is in the classpath where Jira can see it the best locations are usually the `<jira-application-dir>/WEB-INF/classes` or `<jira-application-dir>/WEB-INF/lib` subdirectories of your [Jira installation directory](#) (as JAR files).

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. Select **Advanced > Listener** to open the Listeners page.
3. In the 'Add Listener' form at the bottom of the page, complete the following fields:
  - 'Name' an appropriately descriptive name for the listener.
  - 'Class' the fully-qualified class name of your listener.

**Add Listener**

Add a new listener by entering a name and class below. You can then edit it to set properties.

Name

Class

[> Built-in Listeners](#)

To use one of Jira's built-in listener classes, first click the '**Built-in Listeners**' link to expand the list of listener classes and then click the name of the specific class in the list. The fully-qualified class name of the built-in listener will be added to the 'Class' field.

4. Click the '**Add**' button and the listener will now be added to the list of listeners above.

## Editing listener properties

If your listener accepts parameters or properties, you can edit these by clicking the '**Edit**' link associated with your listener (on the 'Listeners' page in Jira's Administration area).

When defining your own Listener, there is a method `getAcceptedParams` to overload for defining the parameter names which are passed as an array of String objects. The `init` method is given a `Map` with the configured values (the JavaDoc is outdated). The `com.atlassian.jira.event.listeners.DebugParamListener` class is a good example of doing this with two parameters.

## Removing a listener

To remove a listener, click the '**Delete**' link associated with that listener (on the 'Listeners' page in Jira's Administration area).

## Custom events

With the ability to [add custom events](#) to Jira, the listener must be updated to deal with the event as appropriate. This is possible by providing an implementation for the method `customEvent(IssueEvent event)` in the listener. For example, the `MailListener` implementation passes the custom event on for notification processing. The `DebugListener` logs that the custom event has been fired.

## See also

- [Tutorial - Writing Jira event listeners with the atlassian-event library](#) this describes how to write listeners using the Atlassian Events library (see [Jira-specific Atlassian Events](#)), rather than the Jira Listener Events described above.

# Managing webhooks

Webhooks are user-defined HTTP POST callbacks. They provide a lightweight mechanism for letting remote applications receive push notifications from Jira, without requiring polling. For example, you may want any changes in Jira bugs to be pushed to a test management system, so that they can be retested.

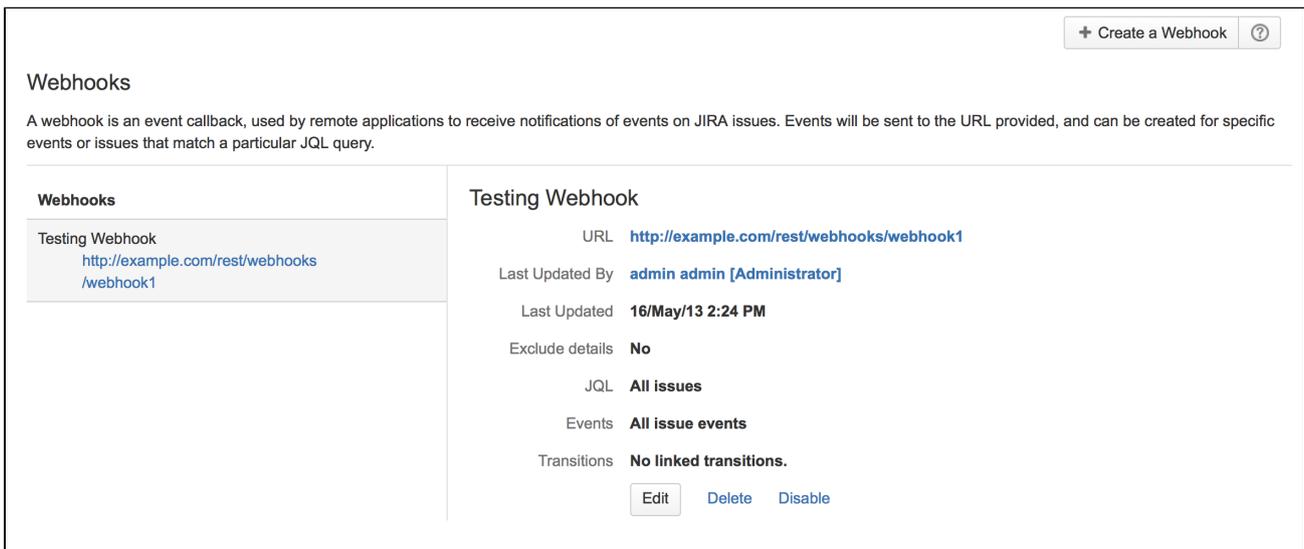
**On this page:**

- [Managing webhooks in Jira](#)

**i** Please read the [Jira Webhooks](#) page (Jira developer documentation) for detailed information on how to configure Jira webhooks, including a description of the events, how to register a webhook via the REST API, examples, and more. This page only contains instructions on how to use the Webhooks user interface in the Jira administration console.

## Managing webhooks in Jira

1. Log in as a user with the **Jira Administrators** global permission.
2. Choose **Administration**  > **System**. Select **Advanced** > **Webhooks** to open the Webhooks page, which shows a list of all existing webhooks.
3. Here's a few tips on using this page:
  - Click the summary of the webhook in the left 'Webhooks' column to display the details of the webhook. You can edit, delete and disable it via the details panel.
  - Deleting a webhook removes it permanently. If you just want to prevent it from firing, disable the webhook instead.



The screenshot shows the Jira administration console for webhooks. At the top right, there is a '+ Create a Webhook' button and a help icon. The main heading is 'Webhooks'. Below it, a descriptive text states: 'A webhook is an event callback, used by remote applications to receive notifications of events on JIRA issues. Events will be sent to the URL provided, and can be created for specific events or issues that match a particular JQL query.'

The interface is split into two columns. The left column, titled 'Webhooks', contains a list with one entry: 'Testing Webhook' with the URL 'http://example.com/rest/webhooks/webhook1/webhook1'. The right column, titled 'Testing Webhook', shows the details for this specific webhook:

- URL: <http://example.com/rest/webhooks/webhook1>
- Last Updated By: [admin admin \[Administrator\]](#)
- Last Updated: **16/May/13 2:24 PM**
- Exclude details: **No**
- JQL: **All issues**
- Events: **All issue events**
- Transitions: **No linked transitions.**

At the bottom of the details panel, there are three buttons: 'Edit', 'Delete', and 'Disable'.

# Services

A service is a class that runs periodically within Jira. Since a service runs inside Jira, it has the ability to use all of the [Jira API](#) and, as it is written in Java, it can use any Java libraries.

Services are useful because they enable you to integrate with external systems by pulling data into Jira periodically. Jira comes with a number of pre-written services, and custom services can be written and plugged in at runtime.

## ✓ Writing a new service?

If you are not extending a built-in Jira service, you should strongly consider writing your new service using the SAL API. Please see our [Scheduling Events via SAL Tutorial](#) for more information.

 For all of the following procedures, you must be logged in as a user with the **Jira System Administrators** [global permission](#).

## Registering a service

 For custom-written services, make sure your service class is in the classpath where Jira can see it the best locations are usually the `<jira-application-dir>/WEB-INF/classes` or `<jira-application-dir>/WEB-INF/lib` subdirectories within of your [Jira application installation directory](#) (as JAR files).

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. Select **Advanced > Services** to open a page showing all the configured services.
3. In the **Add Service** form at the bottom of the page, complete the following fields:
  - **Name** a descriptive name for this service.
  - **Class** the fully-qualified class name of your service. This is likely to have the form `com.atlassian.jira.service.services.type.TypeService`  
See [Sample services](#) for provided service class names.
    -  To use one of Jira's built-in service classes, first click the **Built-in Services** link to expand the list of service classes and then click the name of the specific class in the list. The fully-qualified class name of the built-in service will be added to the **Class** field.
  - **Delay** the delay (in minutes) between service runs.  
For example, to add a debugging service, click the **Built-in Services** link followed by the **Debugging Service** link.
4. After completing the fields on the **Add Service** form, click the **Add Service** button. This opens the **Edit Service** page, where you can configure your new service's options.
  -  Your service's options will vary depending on the type (i.e. class) of service you chose.
5. After completing the remaining options on the **Edit Service** page, click the **Update** button to save your new service's options.

## Editing service properties

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. Select **Advanced > Services** to open a page showing all the configured services.
3. Click the **Edit** link associated with the service whose properties you wish to edit.

For example, to change the interval at which email is sent from Jira, edit the **Mail Queue Service** and change the **Delay** from the default value of 1 minute.

## Removing a service

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. Select **Advanced > Services** to open a page showing all the configured services.
3. Click the **Delete** link associated with the service you wish to remove.

## Built-in services

Jira has some useful services out of the box, which may be used as-is or modified for use in your own environment. The source code for all built-in services is available and should give you a good overview of how simple it is to write your own services. All built-in services are included with Jira and need only be configured to be used.

#### Export service

The Export Service is useful for periodically backing up Jira. It exports all data from Jira every time it is run, into a directory supplied as a parameter. The export files are timestamped, thus the service can act as a backup system.

To test this service, add a service with the class **com.atlassian.jira.service.services.export.ExportService**. Jira sets up an ExportService in new Jira installations (once the setup wizard has been completed). Hence, you may find you already have one.

You can find this class within the following directory of an expanded Jira source archive (which can be downloaded by Jira customers from <https://my.atlassian.com>):

```
<source-installation-directory>/jira-project/jira-components/jira-core/src/main  
/java/com/atlassian/jira/service/services/export
```

#### Mail handler services

Jira mail handler services are not configurable through Jira's **Services** page (with the exception of being able to be removed). For more information about configuring a mail handler in Jira, including the creation of custom mail handlers, please refer to [Creating issues and comments from email](#).

## Custom services

If you are a Jira developer who wishes to write your own Jira service, please note that Jira Service classes must all extend [com.atlassian.jira.service.JiraService](#). Most do so by extending [com.atlassian.jira.service.AbstractService](#) or some more specialized subclass.

# Using AppLinks to link to other applications

Application Links (sometimes called *app links*) is a bundled app that allows you to set up links, share information, and provide access to certain resources or functionality across multiple Atlassian products. We recommend using OAuth authentication for application links because of the greater security inherent with that protocol. We no longer recommend the Trusted Applications and Basic authentication types.

Linking Jira to other applications allows you to include information from these systems in Jira projects and issues. For example, if you link Jira to Confluence, you can include pointers to wiki pages when creating or editing issues. Another common use case is to link Bitbucket Server with Jira; this allows you to view branches, commits and pull requests that correspond to your stories in Jira. In addition to Atlassian applications, you can also link to external applications; for example, you might use an app that allows you to share Zendesk or Salesforce data via an application link.

## Create an application link

1. Log in to Jira as a user with 'Jira Administrator' permissions.
2. Select **Administration**  > **Applications**. Select **Application Links** in the left menu.
3. Enter the URL of the application you want to link to, then select **Create new link**.

### Cloud to Cloud integration

When adding the Cloud URL, use the **https://instance.atlassian.net**. If the **https://** is not used, the link will not be completed.

- If you check **The servers have the same set of users...**, then this link will be configured using OAuth (with impersonation) authentication.
  - If you are *not* an admin on both servers you won't be able to set up a 2-way (reciprocal) application link. If you want to go ahead and create a 1-way link anyway, clear the **I am an administrator on both instances** checkbox.
4. Use the wizard to finish configuring the link. If the application you are linking to does not have the Application Links plugin, you must supply additional information to set up a link with OAuth authentication.

When you complete the wizard, the Application Links plugin will create the link between your applications using the most secure authentication method that is supported between the two applications. See the [Application Links User Guide](#) for more information.

The new link will appear on the Configure Application Links page, where you can:

- Edit the settings of the application link (for example, to [change the authentication type](#) of the link) using the Edit (  ) icon.
- Specify the default instance if you have multiple links to the same type of application (for example, to multiple Jira servers) using the **Make Primary** link. See [Making a primary link for links to the same application type](#) for more information.

## Impersonating and non-impersonating authentication types

### OAuth authentication

OAuth authentication redirects a user to log in to the remote application, after which tokens generated on their behalf are used to authorize requests made from the local application. The remote application handling the request uses the access permissions of the account with which the user logged in on that remote application.

Typical scenarios include:

- You are setting up an application link between two applications that do not share the same set of users.

- You want to continue using a link to an application that now allows public sign-on and the link was previously configured with a shared userbase. You can update your application link by changing **OAuth (impersonation)** to **OAuth** when editing the application link.

See [OAuth security for application links](#) for more information.

### OAuth with impersonation

Atlassian OAuth with impersonation makes it easy for your users to benefit from the deep integrations between Atlassian applications:

- they're automatically authenticated on the other application and don't get asked to authorize requests.
- they'll only see the information that they have permission to see.

Impersonating authentication makes requests on behalf of the user who is currently logged in.

Note that Atlassian OAuth with impersonation can only be used for application links between Atlassian applications. Furthermore, it should only be used when the two applications share the same userbase, typically managed with an external directory using LDAP.

A typical scenario is:

- You've set up an application link but your users still have to authenticate regularly. This can occur when the application link has been configured to not share the same userbase. If those applications do share the same userbase, you can update your application link by selecting **OAuth (impersonation)** when editing the application link.

See [OAuth security for application links](#) for more information.

### Linking to developer tools

When you create a new application link between Jira and an instance of Bitbucket Server, FishEye, Crucible or Bamboo, 2-legged (2LO) and 3-legged OAuth (3LO) are enabled by default. 2LO is required for information from any of those applications to be included in the summaries in the Development panel; 3LO is used to ensure that a user has authenticated with the other applications before they get to see the information in any of the details dialogs.

An existing application link between Jira and Bitbucket Server, FishEye, Crucible or Bamboo (that perhaps used Trusted Apps authentication) needs to have 2-legged authentication (2LO) enabled for both outgoing and incoming authentication, so that information from the application can be included in the Development panel summaries.

When updating an older application link to use OAuth, 3-legged authentication is applied by default, but you need to explicitly enable 2LO. Enable 2-legged authentication for the application link from within Jira as follows:

1. Go to the Jira admin area and click **Applications**.
2. Click **Edit** for the app link with the other application.
3. For both **Outgoing Authentication** and **Incoming Authentication**:
  - a. Click **OAuth**
  - b. Check **Allow 2-legged OAuth**.
  - c. Click **Update**.

The application link update process will involve logging you into the other application for a short time to configure that end of the link, before returning you to Jira.

### Troubleshooting

#### Having trouble integrating your Atlassian products with application links?

We've developed [a guide to troubleshooting application links](#), to help you out. Take a look at it if you need a hand getting around any errors or roadblocks with setting up application links.



# Administering projects and links across multiple applications

The following pages introduce the various aspects of project administration in Jira, Confluence, and Bamboo Cloud. You can also learn more about creating links between Atlassian Server and Cloud applications.

## On this page:

- [Configuring a cross-application link](#)
- [Creating a cross-application project](#)
- [Deleting a cross-application project](#)
- [Linking to server applications from Atlassian Cloud](#)

## Configuring a cross-application link

In order to create links in Jira to point to other applications, the fields where links are created must use the *Wiki Style Renderer*, e.g. the *Comment* and *Description* fields of your project's Jira issues.

## Procedure

1. Choose **Administration**  **>Projects**.
2. Select the project that contains the fields you need to configure.
3. On the Jira project configuration page, go to the **Fields** section.
4. On each field where you will create links, click the **Renderers** link.
5. From the **Active Renderer** drop-down, select **Wiki Style Renderer**.

If the field configuration applies to multiple projects and you don't want other projects to be affected by the change, copy the referenced **Field Configuration Scheme** and associate your project with the new field configuration before changing the settings.

## Creating a cross-application project

If you have multiple Cloud applications, Jira projects can be associated with objects in the other applications, such as:

- awiki space
- abuild project
- asource repository

The association among these objects makes it possible to automatically link issues, wiki documents, plan, and build result.

### Before you begin

Ensure that you at least have access to create projects in Jira. If you have additional access to other applications in your site (such as Confluence and Bamboo), you can automatically create a corresponding Confluence space or a Bamboo project when your Jira project is created. For information on how to set application access, see the documentation on [giving users access to Jira applications](#).

### Creating a project

After you complete the setup wizard, a Jira project is created. If you've chosen to have your site create corresponding objects in other applications, those are created, too. Alternatively, you can manually create objects in other applications and link them together yourself.

After the project is created, you will see the global settings for the project and can continue to configure application-specific settings for the project as needed.

#### Notes:

- Choosing **Jira Classic** or **Project Management** creates the *default Jira project*.
- The *project key* will be used as the prefix of this project's issue keys (e.g. *TEST-100*). Choose one that is descriptive and easy to type.
- The *project lead* is a unique project role. Choose the person who manages the project as the project lead. If there is only one user in your Jira system, the Project Lead defaults to that person and this field is not available.

## Deleting a cross-application project

Deleting a project will delete the project from Jira only. Any wiki documents, source files, changesets, code reviews, build plans, and build results associated with the project will remain in their respective applications. To delete projects, you must have project admin access for the project in Jira.

1. Select **Projects > View all projects**. You will see all your projects on the page.
2. Locate the project you want to delete. Click the **Administer Project** button.
3. From the **Actions** button, click **Delete Project**.

The project will be deleted, but corresponding spaces, changesets, etc. in other applications will remain. You must delete them from those application if you want to remove them.

## Linking to server applications from Atlassian Cloud

You can set up application links between your cloud and server applications. For more information, see [Link to server products from cloud](#).

# Integrating with collaboration tools

## Integrating with Confluence

Give your team the ability to share, discuss and work with Jira application issues in Confluence, and create knowledge articles for your Jira Service Management customers. Here are some of the ways you can benefit from integrating Confluence with your Jira applications:

For...	You can...
<b>Bugs</b>	Create a knowledge base article to document a workaround for a bug.
<b>New Features</b>	Create a product requirements document for a new feature.
<b>Self-service</b>	Create knowledge articles that customers can view on the customer portal to find solutions themselves
<b>General Jira Use Case</b>	Document and collaborate with your team on an issue in Confluence.

And here are just a few of the things Confluence allows you to do:

- Share pages
- Watch pages
- Create knowledge articles from Jira Service Management issues
- Collaborative commenting, especially through the use of @mentions
- Form a team network and let them know what you are doing via a status update
- Add images, picture galleries, videos, and more
- Enable various content macros

See [Integrating Jira and Confluence](#) for more information.

## Integrating with Hipchat

 This page also assumes that you are using the latest version of the integration plugin. Some features may not be available with previous versions of the plugin.

You can connect multiple instances of Jira to the same Hipchat instance, however you can't connect multiple Hipchat instances to the same Jira instance. [Integrating Jira applications and Hipchat](#) gives you and your team the following collaboration power:

- Get notifications in your Hipchat rooms when a customer updates a Jira Service Management request, or a developer comments on an issue.
- Create a dedicated Hipchat room from the issue you're working on and want to discuss with your team.
- Preview Jira issues and Jira Service Management requests directly in Hipchat when someone on your team mentions them.

## Before you begin

The Jira Server and Hipchat integration shares information between the two applications in the following ways:

- **Push:** Jira sends notifications to Hipchat.
- **Pull:** Hipchat retrieves information from Jira. If your Jira server is behind a firewall, you will need to make the server addressable from the internet (by assigning an addressable URL). If you are unable to access your Jira server from behind the firewall, you can still use the integration, you will just be unable to receive pull messages such as Jira Issue Preview. Alternatively, you can install and configure Hipchat Server from behind the same firewall.

Connection status is displayed in the Connect field.

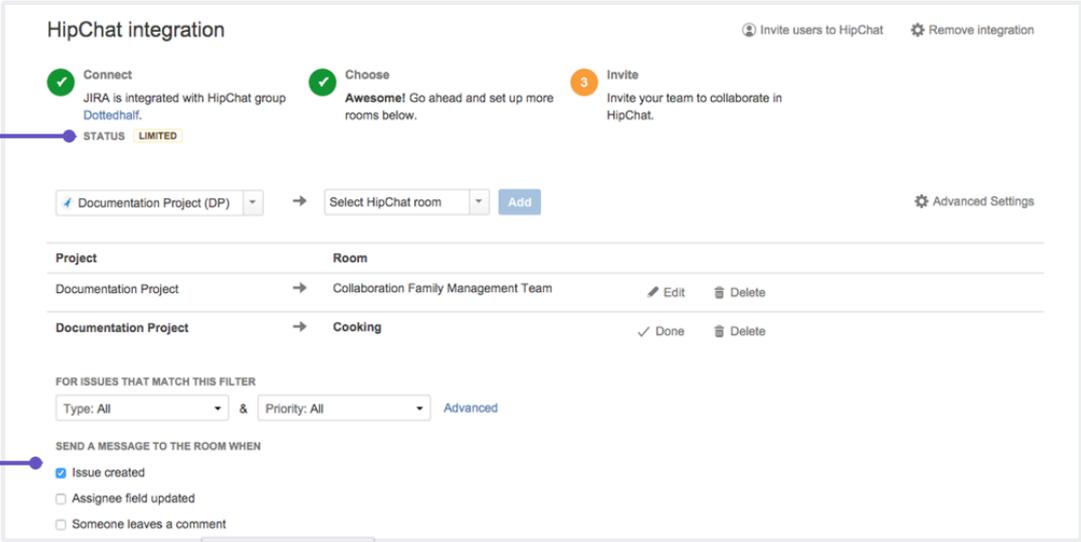
- **Connected:**Hipchat and Jira are connected and working just fine. Carry on.
- **Limited:**Hipchat cannot connect to your Jira server - it may be behind a firewall. You can still receive messages from Jira in Hipchat, but some functionality (such as Issue Preview and @mentions) may not work.
- **Not Connected:**Could not connect to the Hipchat server. Integration features will be unavailable until the connection is restored.To diagnose connection issues, contact your Jira Administrator.
- **Unknown:**Hipchat cannot determine the connection status and may be unable to connect to your Jira server. Some, or all, functionality may not work.

## Linking Jira and Hipchat

1. Log in as a Jira administrator or a Project Administrator.
2. Select **Administration**  **Applications**.
3. Scroll down the page to the **Integrations** section and select **Hipchat**.
4. Select **Connect Hipchat**.
5. Follow the instructions on the screen to link Jira to your Hipchat site.
6. If your Hipchat server runs over HTTPS, the SSL certificate must be imported into Jira's trust store. See [Connecting to SSL services](#) for information on configuration.

## Setting up Jira notifications in Hipchat

1. Sign in to Jira as an administrator.
2. Select **Administration**  **Applications**.
3. Under **Integrations**, select **Hipchat**.
4. Create a link between a project and a Hipchat room:
  - Select a project from the project drop-down menu
  - Select a Hipchat room
  - Click **Add**
5. Alternatively, click **Edit** to change an existing link.
6. Configure the notification settings you'd like to use



**HipChat integration** Invite users to HipChat Remove integration

**1** **Connect**  
 JIRA is integrated with HipChat group Dottedhalf.  
 STATUS **LIMITED**

**2** **Choose**  
 Awesome! Go ahead and set up more rooms below.

**3** **Invite**  
 Invite your team to collaborate in HipChat.

Documentation Project (DP) → Select HipChat room **Add** Advanced Settings

Project	Room		
Documentation Project	→ Collaboration Family Management Team	Edit	Delete
Documentation Project	→ Cooking	✓ Done	Delete

FOR ISSUES THAT MATCH THIS FILTER  
 Type: All & Priority: All Advanced

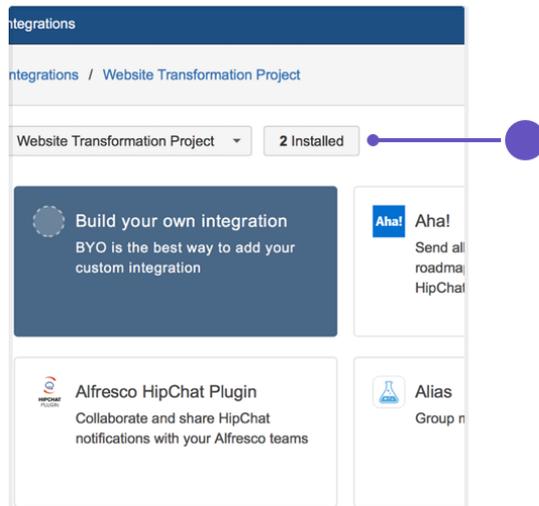
SEND A MESSAGE TO THE ROOM WHEN

- Issue created
- Assignee field updated
- Someone leaves a comment

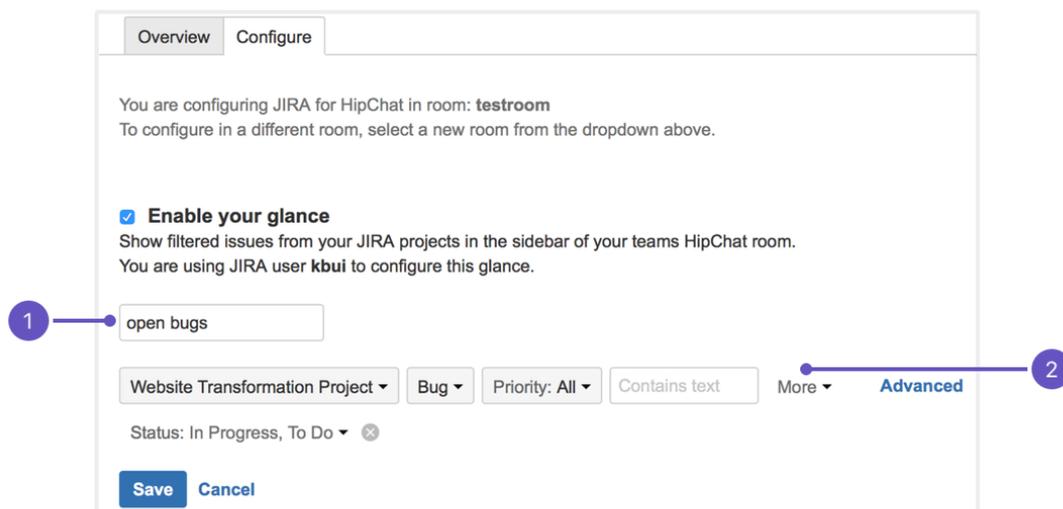
1. **Status:** Connection between Jira and Hipchat.
2. **Messages:** Select messages to send to the room.

## Setting up Jira issues in the Hipchat sidebar

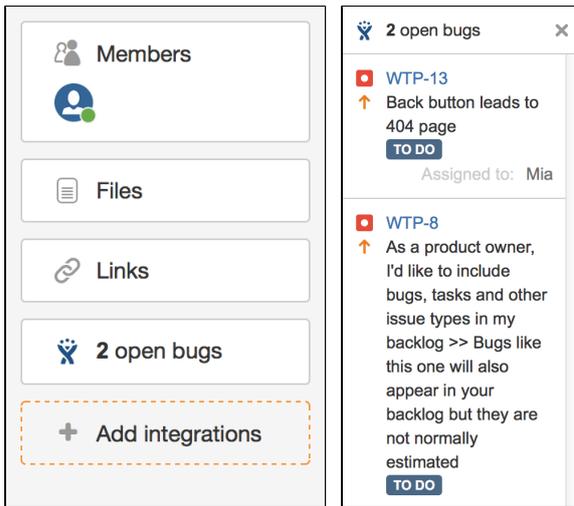
1. Sign in to Hipchat. You'll need to be an administrator of the rooms you want to configure.
2. Select **Integrations** from the top menu.
3. From the drop-down, select the room you'd like to configure.
4. Select **Installed** to see the integrations that have been installed for this room.



5. Select the Jira integration.
6. Select **Enable your glance**. The glance settings will appear.
7. Give your glance a name and set up a basic or JQL filter. The glance name should represent the filter's purpose.



1. **Name:** This will display in the Hipchat room's sidebar above your issues.
  2. **Issue filter:** The issues you want to display in the sidebar.
8. Click **Save**. Open your room in Hipchat and click the glance to see these issues in your sidebar.



## Remove OAuth Permissions

You can remove permissions that you have granted to allow Jira to access Hipchat. For instance, if you have given Jira permission to invite users on Hipchat's behalf.

1. Select your avatar to access your profile.
2. Click **Profile**.
3. Select **Tools**.
4. Click **Hipchat OAuth Sessions**.
5. Select **Remove Access**.

## Integrating with Portfolio for Jira

Portfolio for Jira provides a single, accurate place for viewing, planning and managing your work across multiple teams and projects. See [our guide](#) to how Jira and Portfolio for Jira work together.

# Managing apps

## About apps

An app is an installable component that supplements or enhances the functionality of Jira in some way. For example, the [Jira Calendar Plugin](#) is an app that shows the due dates for issues and versions in calendar format. Other apps are available for connecting Jira to Bamboo, developing for Jira, and accessing Atlassian support from Jira.

Jira comes with many pre-installed apps (called system apps). You can install more apps, either by acquiring an app from the [Atlassian Marketplace](#), or by uploading an app from your file system. This means that you can install apps that you have developed yourself. For information about developing your own apps for Jira, see the [Jira Developer documentation](#).

To enable various Jira Gadgets, see [Configuring the default dashboard](#).

## About the Universal Plugin Manager

The Universal Plugin Manager (UPM) is itself an app that you use to administer apps from the Jira Administration console. UPM works across Atlassian applications, providing a consistent interface for administering apps in Jira, Confluence, Crucible, Fisheye, Bitbucket Server, or Bamboo.

UPM comes pre-installed in recent versions of all Atlassian applications, so you do not normally need to install it yourself. However, like other apps, the UPM software is subject to regular software updates. Before administering apps in Jira, therefore, you should verify your version of the UPM, and update it if needed.

You can update UPM or any app from the UPM's own app administration pages. In addition to updating UPM, you can perform these tasks from the administration pages:

- Install or remove apps
- Configure app settings
- Discover and install new apps from the [Atlassian Marketplace](#)
- [Enable or disable apps](#) and their component modules, including "safe mode"

If the app request feature is enabled in your Atlassian application, non-administrative users can also discover apps in the Atlassian Marketplace. Instead of installing the apps, however, these users have the option of requesting the apps from you, the administrator of the Atlassian application.

For more information on administering the app request feature or performing other common app administration tasks, see the [Universal Plugin Manager documentation](#).

### On this page:

- [About apps](#)
- [About the Universal Plugin Manager](#)

# Upgrading Jira applications

## Reasons to upgrade



A new version means new features, better performance, and continued support to name just a few. Here you can read more about the benefits of upgrading.

[Reasons to upgrade](#)

## Upgrade checklist



Not sure what steps an upgrade involves, or maybe did it a thousand times and just need a quick cheat sheet? Grab this handy checklist for an overview of all steps.

[Upgrade checklist](#)

---

## Pre-upgrade steps

Let's get these steps out of the way to make sure you have a smooth upgrade.

### Testing the upgrade

Create a replica of your Jira instance in a testing environment, so you can test the upgrade.

[Create a test environment](#)

### Preparing for the upgrade

Get Jira ready for the upgrade by running a health check, checking app compatibility, and creating backups.

[View pre-upgrade steps](#)

---

## Upgrade methods

Not sure which one to choose? [Learn more about upgrade methods](#)

### Server or Data Center



Jira Server or Data Center (non-clustered)

#### **Installer**

Better for Windows

#### **Manual**

Better for Linux

#### **Fallback**

Quickly roll back if upgrade fails

### Data Center



Clustered Jira Data Center

#### **Installer**

Better for Windows

#### **Manual**

Better for Linux

#### **Fallback**

Quickly roll back if upgrade fails

#### **Zero downtime**

Keep Jira active during the upgrade

# Reasons to upgrade

## Upgrading from Jira 7.x to Jira 8.x

 If you're already on the Jira 8.x line, feel free to skip this section.

Jira 8.0 was a platform release, and as every platform release it brought significant changes which you wouldn't expect in a smaller, feature release. All of them affect the upgrade in some way, and require that you take under consideration extra steps. We've listed everything you need to know below.

### Reindexing

One of the changes coming with Jira 8.0 is the upgrade of the Lucene library, which is responsible for the index in Jira. This change improves indexing (and searching), but it also makes your current index incompatible with the new version. That's nothing to worry about, as you can rebuild the whole index by reindexing Jira - an action you would probably run after a normal upgrade anyway. After reindexing, the index will be compatible with any later version.

#### What changes

Your current index will remain in `<home-directory>/caches/indexes` (feel free to delete it after the upgrade). From now on, the new index will be stored in the following directory:

```
<home-directory>/caches/indexesV1
```

#### How does it affect the upgrade

The new index will trigger an automatic reindex in Jira just after the upgrade (more precisely, when you start Jira). Because of that, there's a great chance you'll reindex twice - after starting Jira and after making some changes that require reindexing, for example upgrading plugins. Reindexing might be really time-consuming in large Jira instances, that's why we're recommending that you disable the automatic reindex and run it manually later, when you're ready.

#### How to disable automatic reindex

We've included steps needed to disable the reindex in our upgrade procedures. If you're looking for a quick answer, though, you'll need to add the following line to the `<home-directory>/jira-config.properties` file before you start Jira after the upgrade.

```
upgrade.reindex.allowed=false
```

#### Recommended actions

- ✔ Disable automatic reindex to avoid reindexing twice, and run the reindex manually, whenever you're ready.
- 📌 If your Jira instance is small and an extra reindex doesn't sound like a problem, you don't need to disable it. Your upgrade won't be affected in any way; it's just for saving you some time.

#### Disabling incompatible apps (add-ons)

You need to disable apps that are incompatible with Jira 8.0 or upgrade them to compatible versions, if these are available. Incompatible apps might block the upgrade and the Jira startup, because they're using elements that are no longer available or that have changed in the new version, like the Jira index, APIs, or some parts of the UI.

#### Recommended actions

- ✔ Disable all incompatible apps, or upgrade them if they have compatible versions. For more info on what you should do, see [Preparing for the upgrade](#).

🔗 We always recommend that you test the upgrade in a staging environment. Once you upgrade the staging environment to Jira 8.0, you can try enabling the incompatible apps and test how they behave with the new version. If they don't affect your Jira instance in a significant way, you can use them with 8.0 even if they haven't been declared as compatible yet.

## Deciding to upgrade

Upgrading is a complex decision to make. Naturally, there is significant time investment you need to consider. On the other hand there are new features, performance improvements, bug fixes, continued support access, company requirements, and maintaining the latest software versioning for your teams.

### Upgrade - what am I getting

#### New features

For every new version of Jira Software, we publish Jira Software release notes and Jira Software upgrade notes. These resources have important information relating to your upgrade, so we recommend reading all of them.

Our upgrade matrix lists all the features and improvements introduced in consecutive Jira releases. The matrix lets you compare versions and select the one you want to upgrade to. It also gives you an overview of bugs fixed in each release.

If you want to dig deeper and learn about all the technical elements that are specific to the Jira Software upgrade process for that version, check the "Technical upgrade notes section" in the matrix.

[View upgrade matrix](#)

#### Performance improvements

Many of our releases focus on performance improvements so you and your team can have an effective product experience with minimal technical slowdown. Two examples of these improvements are optimization of indexing time and faster page load time for dashboards and issues.

[View performance improvements](#)

#### Bug fixes

We use this public Jira instance to track the development of product bug fixes and suggestions. Many customers locate issues and suggestions that would benefit their team and vote or watch them. Just like watching Jira issues in your instance, you will be notified via email when an issue is updated.

[View recently resolved issues](#)

[View new features policy](#)

#### Access to support

For Long Term Support releases, we provide support and bug fixes for all server products for 2 years after a version is released. Ending support after 2 years allows us to focus our resources on supporting more modern versions used by the majority of customers. This means that after a version has reached its end of life, our support team will only be able to assist with upgrade-related questions.

For versions, which are not Long Term Support releases, we provide support for 2 years, but the fixes are available only until the next version is out. Then, we only fix critical bugs. If you want to get all the new features and fixes, it's good to upgrade versions.

[View End of life policy](#)

### Upgrade - what I need to consider

In addition to the technical considerations, there are a few other factors to consider that can help ensure a smooth upgrade.

#### Time

The amount of time required depends on several factors, including the size of your instance, number of apps, and amount of customizations. As a general rule, it's best to incorporate buffer time for planning and execution in case anything unexpected comes up.

#### Tools

There are no external tools required to complete a Jira upgrade. However, there are a few handy resources that can be used to help you along the way, like the [app compatibility checker](#) and the Instance Health Check that you can find by going to **Administration** (⚙️) > **System** > **Troubleshooting and support tools**.

#### Stakeholders

While some people might wear many hats in your organization, generally it's a good idea to carefully consider how to communicate information about the upgrade, especially if it will introduce major changes. You might consider informing an approver, end users, and, if you've acquired your license through an approved Atlassian Solution Partner, your partner as they may offer additional services for assisting you during the upgrade process.



All ready?

Head back to the upgrade matrix, and choose the version you want to upgrade to.

[Go to the upgrade matrix](#)

# Upgrade checklist

Use this checklist as a guide to ensure you're set up for success for your Jira Software upgrade. Keep in mind that every instance is unique, and this guide should be supplemented with tasks and customizations specific to your instance.

You can use this template and **make additions to suit your Jira Software instance**.

[View checklist as a Trello board](#)

Stage	Task	Notes	Status /Comment
Research & decision	Decide which version to upgrade to	<ul style="list-style-type: none"> <li>Use the <a href="#">Jira Software upgrade matrix</a> to compare versions and select the version that suits your needs. Then go to <a href="#">upgrade notes</a> to check the technical aspect of the upgrade.</li> </ul> <div style="border: 1px solid green; padding: 5px; margin-top: 10px;">  <b>Pro Tip:</b> if you upgrade infrequently, a Long Term Support latest release might be the version for you.         </div>	
	Outline upgrade stakeholders (optional)	<ul style="list-style-type: none"> <li>List roles, responsibilities, and contact info to keep the interested parties in the loop and be aware of all the dependencies.</li> </ul>	
Preparation & testing	Check maintenance status	<ul style="list-style-type: none"> <li>You need to have a valid licence to upgrade Jira. Check and renew at <a href="https://my.atlassian.com">my.atlassian.com</a>.</li> </ul>	
	Get your list of files with custom modifications	<p>Have you modified files to customize your Jira? If you want to keep these changes, make sure you know which files have been modified.</p> <ul style="list-style-type: none"> <li>If you're on the latest version of the <a href="#">ASTS plugin</a>, go to <b>Jira administration</b>  <b>Applications &gt; Plan your upgrade</b> to see the list of files in which you introduced custom changes.</li> </ul> <p>If you've made changes to</p> <ul style="list-style-type: none"> <li>- &lt;jira-home-directory&gt;/atlassian-jira/ directory</li> <li>-&lt;jira-home-directory&gt;/conf/server.xml</li> <li>-&lt;jira-home-directory&gt;/bin/setenv.sh</li> </ul> <p>we'll automatically transfer these changes to your upgraded Jira before you start it. However, if you've made changes to other files, you'll need to manually transfer them.</p> <ul style="list-style-type: none"> <li>If you can't pull up the <b>Plan your upgrade</b> page, compile the list of files you've modified. For some suggestions regarding files that get modified most frequently, see <a href="#">here</a>.</li> </ul> <p>For more info, see <a href="#">Pre-upgrade planning tool</a>.</p>	
	Check supported platforms	<ul style="list-style-type: none"> <li>It might happen that some platforms reached their end of life and we stopped supporting them. Check the <a href="#">Supported Platforms</a> page for details.</li> </ul>	

<p>Set up your testing environment</p>	<ul style="list-style-type: none"> <li>• Check your upgrade on a test environment first. Remember that your test environment must be a copy of your working environment. See <a href="#">Creating a test environment for Jira</a></li> <li>• If necessary, adjust memory or reverse proxy settings based on the <a href="#">upgrade notes</a>.</li> <li>• For resourcing guidelines, check out the <a href="#">Jira sizing guide</a>.</li> </ul>	
<p>Run an instance health check</p>	<ul style="list-style-type: none"> <li>• See if your instance is good to upgrade. Health check is embedded in the Support Tools Plugin. More details <a href="#">here</a>.</li> </ul>	
<p>Check the compatibility of your apps</p>	<ol style="list-style-type: none"> <li>1. Check app compatibility with the <a href="#">Jira update check</a>.</li> <li>2. Because of major changes we've introduced in Jira 8.0, you need to disable all incompatible apps before the upgrade. We also recommend disabling the apps with the status: <ul style="list-style-type: none"> <li>• <b>Incompatible</b></li> <li>• <b>Compatible if both upgraded</b></li> <li>• <b>Unknown</b></li> </ul> </li> <li>3. Upgrade your apps with the <b>Compatible if upgraded</b> status.</li> </ol> <p>See <a href="#">Incompatible apps</a>.</p>	
<p>Determine your optimal upgrade method</p>	<p>Choose a Jira upgrade method:</p> <ul style="list-style-type: none"> <li>• <a href="#">Upgrading Jira applications using a rapid upgrade method</a></li> <li>• <a href="#">Upgrading Jira Server (manual)</a></li> <li>• <a href="#">Upgrading Jira Server with a fallback</a></li> <li>• <a href="#">Upgrading Jira Data Center (installer)</a></li> <li>• <a href="#">Upgrading Jira Data Center (manual)</a></li> <li>• <a href="#">Upgrading Jira Data Center with a fallback</a></li> <li>• <a href="#">Upgrading Jira Data Center with ZDU</a></li> </ul> <p>If you're starting from Jira 6.4 or earlier, you need to upgrade to Jira 7.x (for example: 7.13) first before you upgrade to 8.x.</p> <div style="border: 1px solid green; padding: 5px; margin-top: 10px;"> <p> <b>Pro Tip:</b> Depending on your OS or whether you're running a DC or a standalone Jira Server, there might be some restrictions to the upgrade method you can choose. See <a href="#">Upgrade methods</a>.</p> </div>	
<p>Do the upgrade</p>	<p><b>CLUSTER</b> When you upgrade, you perform all the pre-upgrade steps, next you perform an upgrade and do the post-upgrade task on one node. Next you create a template and upgrade all other nodes. Only then can you make post-upgrade steps for the entire DC.</p>	

<p>Re-apply any modifications and increase pool-max-size</p>	<p>If you've made changes any of the files copy them over to your upgraded instance. These changes might include:</p> <ul style="list-style-type: none"> <li>• customisations to xml configuration files (e.g so they can use load balancing)</li> <li>• custom icons (e.g replacing the Jira logo with their company logo)</li> <li>• custom email templates (similarly related to the branding thing)</li> </ul> <p>Test customizations in your testing environment prior to upgrading your production instance, because there is a chance that changes during the upgrade will make your Jira Software customizations unusable.</p> <p>If you're unsure which files you might have modified, go to <b>Jira administration</b>  <b>Applications &gt; Plan your upgrade</b> to see the list of files in which you introduced custom changes.</p> <p>Alternatively, take a look at the list of <a href="#">important files in the Jira installation directory</a>. Also, you might want to look in the logs for the <b>Modifications</b> section:</p> <div data-bbox="497 779 1305 945" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p style="text-align: center;">___ Modifications ___</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">Modified Files</td> <td style="padding: 2px; text-align: right;">: None</td> </tr> <tr> <td style="padding: 2px;">Removed Files</td> <td style="padding: 2px; text-align: right;">: None</td> </tr> </table> </div> <p>If you use the installer method, then the following parameters were migrated from your existing Jira Software installation:</p> <ul style="list-style-type: none"> <li>• TCP values in the <code>server.xml</code> file</li> <li>• Location of your Jira home directory in the <code>Jira-application.properties</code> file</li> <li>• The following values in the <code>setenv.sh/setenv.bat</code> file: <ul style="list-style-type: none"> <li>• <code>JVM_SUPPORT_RECOMMENDED_ARGS</code></li> <li>• <code>JVM_MINIMUM_MEMORY</code></li> <li>• <code>JVM_MAXIMUM_MEMORY</code></li> <li>• <code>Jira_MAX_PERM_SIZE</code></li> </ul> </li> </ul> <p>When you know which files have been modified, copy these modifications to the respective files in your upgraded Jira. Copy only the modified parts not the entire files!</p> <p>If you're upgrading to Jira 8.6 and later and running the ATST version 1.20.0 or later, we will show you the files whose modifications have not been copied over on Jira startup. Then, you'll be able to copy the changes automatically.</p> <p>You'll be prompted to restart Jira after copying is completed.</p> <p>If you're upgrading from Jira 7.x to Jira 8.x we recommend changing the <code>pool-max-size</code> parameter to 40 in your <code>dbconfig.xml</code> before the upgrade. Leaving the default of 20 can sometimes lead to ResultSet Closed errors during re-indexing on 8.x. For information on implementing the change, see <a href="#">Tuning database connections</a></p> <p>Any changes to <code>dbconfig.xml</code> require a restart.</p>	Modified Files	: None	Removed Files	: None	
Modified Files	: None					
Removed Files	: None					
<p>Decide when to reindex</p>	<p><b>7.X TO 8.X UPGRADE</b> Jira will automatically remove the old incompatible index and start a full re-index on startup. You might want to postpone reindexing to upgrade your apps first, as some apps will require an additional reindexing. For more information, see <a href="#">Disabling automatic reindexing</a>.</p>					

	Start Jira		
	Upgrade your apps	Now you can upgrade the apps with the <b>Compatible if both upgraded status</b> .	
	Re-apply any modifications (if you haven't done it already)	<p>If you're upgrading to Jira 8.6 and later and running the ATST version 1.20.0 or later, you can see a list the files whose modifications have not been copied over on Jira startup. Then, you can just select to copy the changes automatically.</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>i</b> Note that when checking for changes, we're only looking at the files/folders below:</p> <ul style="list-style-type: none"> <li>- &lt;jira-home-directory&gt;/atlassian-jira/ directory</li> <li>- &lt;jira-home-directory&gt;/conf/server.xml</li> <li>- &lt;jira-home-directory&gt;/bin/setenv.sh</li> </ul> <p>To automatically transfer the changes, the installer copy of the modified file needs to be the same as in the version you're upgrading to.</p> </div> <p>You'll be prompted to restart Jira after copying is completed.</p>	
	Test the upgrade with users	<ul style="list-style-type: none"> <li>• Ask your Jira users if the upgrade works fine for them. This is super helpful if you're passing a major version, ie. 8.0. Select a diverse sample group that utilizes different features of Jira Software.</li> </ul>	
	Document learnings (optional)	<ul style="list-style-type: none"> <li>• Document the things you notices during the upgrade - that can help you with future upgrades.</li> <li>• Update timeline and stakeholders if necessary.</li> </ul>	
	Create a timeline and communicate	<ul style="list-style-type: none"> <li>• Remember to book enough time for your upgrade. Add some buffer time, too.</li> <li>• Communicate the change to end users.</li> </ul>	
<b>Preparation &amp; execution</b>	Check maintenance status	<ul style="list-style-type: none"> <li>• You need to have a valid licence to upgrade Jira. Check and renew <a href="https://my.atlassian.com">atmy.atlassian.com</a>.</li> </ul>	
	Check supported platforms	<ul style="list-style-type: none"> <li>• It might happen that some platforms reached their end of life and we stopped supporting them. Check the <a href="#">Supported Platforms</a> page for details.</li> </ul>	
	Run an instance health check	<ul style="list-style-type: none"> <li>• See if your instance is good to upgrade. Health check is embedded in the Support Tools Plugin. More details <a href="#">here</a>.</li> </ul>	

Check the compatibility of apps	<ol style="list-style-type: none"> <li>1. Check app compatibility with the <a href="#">Jira update check</a>.</li> <li>2. <b>7.X TO 8.X UPGRADE</b> Because of major changes we've introduced in Jira 8.0, you need to disable all incompatible apps before the upgrade. We also recommend disabling the apps with the status: <ul style="list-style-type: none"> <li>• <b>Incompatible</b></li> <li>• <b>Compatible if both upgraded</b></li> <li>• <b>Unknown</b></li> </ul> </li> <li>3. Upgrade your apps with the <b>Compatible if upgraded</b> status.</li> </ol> <p>See <a href="#">Incompatible apps</a>.</p>	
Back up your instance data	<ul style="list-style-type: none"> <li>• Use native database backup tools, unless you're changing some parts of your environment (e.g. database software), in which case you'll need to use a Jira's XML backup utility.</li> <li>• Check out a detailed breakdown of the two processes <a href="#">here</a>.</li> </ul>	
Back up your directories	<ul style="list-style-type: none"> <li>• Back up your installation and home directory. More details <a href="#">here</a>.</li> </ul> <p><b>CLUSTER</b> Remember to do the backup for all the nodes and to also back up the shared directory.</p>	
<b>7.X TO 8.X UPGRADE</b> Decide when to reindex	<p>Jira 8.0 will automatically remove the old incompatible index and start a full re-index on startup. You might want to postpone reindexing to upgrade your apps first, as some apps will require an additional reindexing. For more information, see <a href="#">Disabling automatic reindexing</a>.</p>	
Execute upgrade in production environment	<p>You have the following upgrade methods:</p> <ul style="list-style-type: none"> <li>• <a href="#">Upgrading Jira applications using a rapid upgrade method</a></li> <li>• <a href="#">Upgrading Jira applications manually</a></li> <li>• <a href="#">Upgrading Jira Server with a fallback</a></li> <li>• <a href="#">Upgrading Jira Data Center (installer)</a></li> <li>• <a href="#">Upgrading Jira Data Center</a></li> <li>• <a href="#">Upgrading Jira Data Center with a fallback</a></li> <li>• <a href="#">Upgrading Jira Data Center with ZDU</a></li> </ul> <p>If you're starting from Jira 6.4 or earlier, you need to upgrade to Jira 7.x first before you upgrade to 8.x.</p> <div style="border: 1px solid green; padding: 5px; margin: 10px 0;"> <p> <b>Pro Tip:</b> Depending on your OS or whether you are running a DC or a standalone Jira Server, there might be some restrictions to the upgrade method you can choose. See <a href="#">Upgrade methods</a>.</p> </div> <p><b>CLUSTER</b> When you upgrade, you perform all the pre-upgrade steps, next you perform an upgrade and do the post-upgrade task on one node. Next you create a template and upgrade all other nodes. Only then can you make post-upgrade steps for the entire DC.</p>	
Upgrade the database driver (if using Oracle or MySQL)	<p>To start Jira successfully:</p> <ul style="list-style-type: none"> <li>• Download the latest JDBC driver (<a href="#">Oracle</a> or <a href="#">MySQL</a>)</li> <li>• Place it in <code>&lt;Jira-installation-directory&gt;/lib</code>.</li> </ul>	

<p>Re-apply any modifications</p>	<p>If you have made changes any of the files copy them over to your upgraded instance. These changes might include:</p> <ul style="list-style-type: none"> <li>• customisations to xml configuration files (e.g so they can use load balancing)</li> <li>• custom icons (e.g replacing the Jira logo with their company logo)</li> <li>• custom email templates (similarly related to the branding thing)</li> </ul> <p>If you're unsure which files you might have modified, go to <b>Jira administration</b>  <b>Applications &gt; Plan your upgrade</b> to see the list of files in which you introduced custom changes.</p> <p>Alternatively, take a look at the list of <a href="#">important files in the Jira installation directory</a>. Also, you might want to look in the logs for the <b>Modifications</b> section:</p> <div data-bbox="518 660 1273 788" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <pre> __ Modifications __ Modified Files           : None Removed Files           : None                     </pre> </div> <p>If you use the installer method, then the following parameters were migrated from your existing Jira Software installation:</p> <ul style="list-style-type: none"> <li>• TCP values in the <code>server.xml</code> file</li> <li>• Location of your Jira home directory in the <code>Jira-application.properties</code> file</li> <li>• The following values in the <code>setenv.sh/setenv.bat</code> file:             <ul style="list-style-type: none"> <li>• <code>JVM_SUPPORT_RECOMMENDED_ARGS</code></li> <li>• <code>JVM_MINIMUM_MEMORY</code></li> <li>• <code>JVM_MAXIMUM_MEMORY</code></li> <li>• <code>Jira_MAX_PERM_SIZE</code></li> </ul> </li> </ul> <p>When you know which files have been modified, copy these modifications to the respective files in your upgraded Jira. Copy only the modified parts not the entire files!</p> <p>If you're upgrading to Jira 8.6 and later and running the ATST version 1.20.0 or later, we will show you the files whose modifications have not been copied over on Jira startup. Then, you'll be able to copy the changes automatically.</p> <p>You'll be prompted to restart Jira after copying is completed.</p>	
<p><b>7.X TO 8.X UPGRADE</b> Decide when to reindex</p>	<p>Jira 8.0 will automatically remove the old incompatible index and start a full re-index on startup. You might want to postpone reindexing to upgrade your apps first, as some apps will require an additional reindexing. For more information, see <a href="#">Disabling automatic reindexing</a> .</p>	
<p>Start Jira</p>		
<p>Upgrade your apps</p>	<p>Now you can upgrade the apps with the <b>Compatible if both upgraded status</b>.</p>	

	<p>Re-apply any modifications (if you haven't done it already)</p>	<p>If you're upgrading to Jira 8.6 and later and running the ATST version 1.20.0 or later, you can see a list the files whose modifications have not been copied over on Jira startup. Then, you can just select to copy the changes automatically.</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>i</b> Note that when checking for changes, we're only looking at the files/folders below:</p> <ul style="list-style-type: none"> <li>- &lt;jira-home-directory&gt;/atlassian-jira/ directory</li> <li>- &lt;jira-home-directory&gt;/conf/server.xml</li> <li>- &lt;jira-home-directory&gt;/bin/setenv.sh</li> </ul> <p>To automatically transfer the changes, the installer copy of the modified file needs to be the same as in the version you're upgrading to.</p> </div> <p>You'll be prompted to restart Jira after copying is completed.</p>	
<p><b>Post-upgrade</b></p>	<p>Test the upgrade with users</p>	<ul style="list-style-type: none"> <li>• Asking users if the upgrade works fine for them is super helpful.</li> <li>• Use the same sample group as the test upgrade</li> <li>• Resolve any issues before marking the upgrade as complete</li> </ul>	
	<p>Communicate to end users</p>	<p>Include major feature additions and contact information for questions</p>	
	<p>Complete an upgrade retrospective (optional)</p>	<p>Document what went well, what didn't, and what to change for next time. This will help your next upgrade run even smoother.</p>	

## Resources

Caught between a rock and a hard place? Here are a few resources to consult if you have questions. We recommend proceeding in this order:

- Browse or search the [Jira Software documentation](#) site and the [Jira Software Server Knowledge Base](#). (Be sure to select the appropriate version in the dropdown.)
- Search our [Atlassian Community site](#) for articles and discussions pertaining to your topic. Or, ask a question to tap into Community experts both inside and outside of Atlassian.
- Create an issue at our [support site](#) so that our support engineers can assist you.
- If you purchased your license through an authorized Solution Partner, consult them to learn more about troubleshooting your instance.
- Looking for more assistance? We also offer Premier Support, which includes 24 x 7 support availability, health checks, dedicated senior support engineers, and more. Check out our [Premium Support offerings](#) for more information.



All good?

[Go to landing page](#)

# Upgrade matrix

This upgrade matrix should serve as a starting point when researching which version of Jira Software and Jira Service Management is right for your team.

Here you'll find all of the supported versions including:

- [Jira Software top features](#)
- [Jira Service Management top features](#)
- [Supported platform changes](#)
- [Technical upgrade notes](#)
- [Reference: Performance and scale testing](#)



## Long Term Support releases

If you're only upgrading about once a year, Long Term Support releases are a good choice.

[Tell me more](#)

## Jira Software top features

Here's a summary of the great new features awaiting your users.

8.20	 <b>This is a Long Term Support release.</b> <ul style="list-style-type: none"><li>• No new features to keep everything smooth and stable</li></ul>	<a href="#">Release notes</a>
8.19	<ul style="list-style-type: none"><li>• New wizard for creating plans in Advanced Roadmaps (Data Center)</li><li>• Dependency report in Advanced Roadmaps (Data Center)</li><li>• Filtering epics and versions in backlogs</li><li>• Data pipeline improvements (Data Center)</li><li>• Pagination and UI improvements on Screens</li></ul>	<a href="#">Release notes</a>
8.18	<ul style="list-style-type: none"><li>▪ Comment reactions (Data Center)</li><li>▪ Analyzing how your issues change over time with data pipeline (Data Center)</li></ul>	<a href="#">Release notes</a>
8.17	<ul style="list-style-type: none"><li>• Keep consistent naming of sprints and epics between Jira and Agile at Scale frameworks with Flexible Terminology</li><li>• Refreshed commenting experience with improved appearance, keyboard navigation and support for assistive technologies</li><li>• Custom fields are now included in data pipeline exports (Data Center)</li><li>• Around 100 security vulnerabilities fixed in the Jira platform</li></ul>	<a href="#">Release notes</a>

8.16	<ul style="list-style-type: none"> <li>• Get more insights into the usage of custom fields (Data Center)</li> <li>• Bulk delete custom fields (Data Center)</li> <li>• Default values for the Description field (Data Center)</li> <li>• Let users log in with multiple identity providers (Data Center)</li> <li>• Check the status of your data pipeline exports (Data Center)</li> <li>• Keep Jira safe by disabling basic authentication</li> <li>• Improvements to DVCS accounts</li> </ul>	<a href="#">Release notes</a>
8.15	<ul style="list-style-type: none"> <li>▪ Advanced Roadmaps is now part of Jira Software (Data Center)</li> <li>▪ Displaying image attachments in email notifications</li> </ul>	<a href="#">Release notes</a>
8.14	<ul style="list-style-type: none"> <li>▪ Email templates made better (Data Center)</li> <li>▪ Quickly view issues belonging to an epic</li> <li>▪ DVCS Connector improvements</li> <li>▪ Personal access tokens</li> <li>▪ Embedded Crowd and password encryption</li> <li>▪ Accessibility improvements when creating issues</li> <li>▪ Improved data insights (Data Center)</li> <li>▪ Jira Cloud Migration Assistant bundled with Jira</li> <li>▪ Choosing Bamboo plan branches in Jira</li> </ul>	<a href="#">Release notes</a>
8.13	<p> <b>This is a Long Term Support release.</b></p> <ul style="list-style-type: none"> <li>▪ No new features to keep everything smooth and stable</li> </ul>	<a href="#">Release notes</a>
8.12	<ul style="list-style-type: none"> <li>• More control over your Advanced Audit log</li> <li>• Support for MySQL 8.0</li> <li>• Improved user picker</li> <li>• Less app impact on indexing</li> <li>• Users created automatically with Just-in-time user provisioning</li> </ul>	<a href="#">Release notes</a>
8.11	<ul style="list-style-type: none"> <li>▪ Managing private filters and dashboards</li> <li>▪ Issue detail view can be hidden per board</li> <li>▪ Improved email notifications for mentions</li> <li>▪ Embedded Crowd upgrade</li> <li>▪ Max timeout for the Favorite Filters gadget</li> <li>▪ Restricting sprint selection when creating or editing issues</li> <li>▪ Configuring how fast stale nodes are moved offline (Data Center)</li> </ul>	<a href="#">Release notes</a>
8.10	<ul style="list-style-type: none"> <li>▪ OAuth 2.0 support for incoming mail</li> <li>▪ User anonymization (GDPR) improvements</li> <li>▪ More insight into your custom fields (Data Center)</li> <li>▪ Stale nodes automatically removed (Data Center)</li> <li>▪ Optimized custom fields (Data Center)</li> </ul>	<a href="#">Release notes</a>
8.9	<ul style="list-style-type: none"> <li>• Node and application status for Jira Data Center cluster monitoring</li> <li>• Refreshed Velocity Chart</li> <li>• Accessibility: Text spacing</li> </ul>	<a href="#">Release notes</a>

8.8	<ul style="list-style-type: none"> <li>■ Revamped audit log</li> <li>■ Setting dates for future sprints</li> <li>■ Accessibility settings</li> <li>■ Running Jira Data Center on a single node (available for all versions)</li> </ul>	<a href="#">Release notes</a>
8.7	<ul style="list-style-type: none"> <li>• Anonymizing users for GDPR compliance</li> <li>• PostgreSQL 11 support</li> <li>• OpenID Connect comes to Jira Data Center</li> </ul>	<a href="#">Release notes</a>
8.6	<ul style="list-style-type: none"> <li>• Jira copies over changes files on upgrade</li> <li>• New JVM code cache check</li> <li>• Replying to JIRA notifications in Outlook made way better</li> <li>• Users and roles made better</li> <li>• PostgreSQL 10 comes to Jira</li> <li>• Several older platforms get deprecated</li> <li>• Prefix and suffix search</li> <li>• Accessible dropdown menus</li> <li>• Configurable scheme parameters in Jira REST API for projects creation</li> <li>• Burnup charts in Jira Software</li>   <li>• Self-protect your Jira Data Center and sleep easy with rate limiting</li> <li>• New information in the audit log for Jira Data Center</li> <li>• Cluster monitoring in Jira Data Center</li> </ul>	<a href="#">Release notes</a>
8.5	<p> <b>This is a Long Term Support release.</b></p> <ul style="list-style-type: none"> <li>• Jira Server mobile app distributable to managed devices with MDM</li> <li>• New JVM check available</li> </ul>	<a href="#">Release notes</a>
8.4	<ul style="list-style-type: none"> <li>• Filters in 'Multi user picker' custom field</li> <li>• Batching emails as a default setting</li> <li>• External links open in a new tab</li> <li>• Filters to browse archived issues</li> <li>• Jira becomes more accessible</li> <li>• Lots of versions OK for boards</li> <li>• Time tracking in bulk edit</li> <li>• Templates for deploying Jira in a Docker container (8.0 and later)</li> </ul>	<a href="#">Release notes</a>
8.3	<ul style="list-style-type: none"> <li>• Jira Server mobile app</li> <li>• Content Delivery Network (CDN) for Jira Data Center</li> <li>• AdoptOpenJDK bundled with Jira</li> <li>• List of custom changes after upgrade</li> <li>• New filters to search for custom fields</li> <li>• Browsing projects is faster</li> <li>• Improvements to batching emails</li> <li>• Cluster lock mechanism improved in Data Center</li> <li>• Re-indexing made better</li> </ul>	<a href="#">Release notes</a>

8.2	<ul style="list-style-type: none"> <li>• First milestone in filters on the Archived issues export page (Jira Data Center)</li> <li>• Support for Microsoft SQL Server 2017</li> <li>• Official support for Java 11</li> <li>• AdoptOpenJDK 8 bundled with Jira</li> </ul>	<a href="#">Release notes</a>
8.1	<ul style="list-style-type: none"> <li>• Archiving issues, first milestone (Jira Data Center)</li> <li>• Archiving components</li> <li>• Development info on issue cards</li> <li>• Improvements to boards: resizable Detail View and horizontal scrollbar for boards with many columns</li> <li>• Performance improvements for boards and backlogs, and other pages that use the side navigation</li> <li>• Improvements to deployment templates for Jira Data Center on Microsoft Azure</li> <li>• New JMX metrics for more insights into Jira performance</li> <li>• Sorting projects by columns</li> <li>• Ability to add additional custom fields to batched email notifications</li> <li>• Information about recent activity on user profiles</li> <li>• New avatars</li> </ul>	<a href="#">Release notes</a>
8.0	<ul style="list-style-type: none"> <li>• Jira Software mobile app (beta)</li> <li>• Batching email notifications to reduce the number of emails you're getting from Jira</li> <li>• New look and feel of Scrum/Kanban boards and backlogs</li> <li>• Performance improvements for backlogs</li> <li>• New priority icons</li> <li>• New JQL options: finding authors of updates, and finding link types</li> <li>• New configuration for MySQL 5.7 that allows 4-byte characters</li> <li>• Massive improvements to indexing</li> <li>• REST API for issue type schemes</li> </ul>	<a href="#">Release notes</a>
7.13	 <b>This is a Long Term Support release.</b> <ul style="list-style-type: none"> <li>• Support for OpenJDK 8.</li> </ul>	<a href="#">Release notes</a>
7.12	<ul style="list-style-type: none"> <li>• Custom fields optimizer (Data Center)</li> <li>• Sharing edit rights for filters and dashboards</li> <li>• New events in the audit log (filters and dashboards)</li> <li>• New look and feel of the custom fields page</li> <li>• Disabling <i>Days in column</i> time indicator to improve performance</li> <li>• New compression method (Google's Snappy) when copying the index between the nodes (Data Center)</li> </ul>	<a href="#">Release notes</a>
7.11	<ul style="list-style-type: none"> <li>• Support for IPv6</li> <li>• New columns on the Projects page (<i>Issues</i>, <i>Last issue update</i>)</li> <li>• Development information available in Basic Search</li> <li>• Apache Tomcat 8.5.29</li> </ul>	<a href="#">Release notes</a>

7.10	<ul style="list-style-type: none"> <li>• New look and feel for Jira, including the first wave of changes to the most frequently used pages.</li> <li>• Archiving inactive or completed projects (Data Center).</li> <li>• New events in the audit log.</li> <li>• Case-insensitive labels.</li> <li>• Quick start guide for Jira Data Center.</li> <li>• Sending Jira notifications to Stride rooms and conversations.</li> </ul>	<a href="#">Release notes</a>
7.9	<ul style="list-style-type: none"> <li>• Improved Kanban boards showing only recently modified issues in the Done column.</li> <li>• Searching through versions with a wildcard.</li> <li>• Choosing delimiters when exporting your issues to CSV.</li> <li>• Disabling empty JQL queries to avoid performance issues with empty filters.</li> <li>• New events in the audit log.</li> <li>• Asynchronous cache replication in Jira Data Center.</li> <li>• IPv6 support for MySQL databases.</li> </ul>	<a href="#">Release notes</a>
7.8	<ul style="list-style-type: none"> <li>• Quick search that shows instant results, and displays your recent work if you're not searching.</li> <li>• Dutch language pack.</li> </ul>	<a href="#">Release notes</a>
7.7	<ul style="list-style-type: none"> <li>• REST API to manage priority schemes outside of the user interface</li> <li>• Italian and Finnish language packs</li> </ul>	<a href="#">Release notes</a>
7.6	<p> <b>This is a Long Term Support release.</b></p> <ul style="list-style-type: none"> <li>• Project-specific priorities.</li> <li>• Live monitoring of your Jira instance with JMX.</li> <li>• Subtasks drag 'n' drop.</li> <li>• X-Frame-Options to prevent clickjacking.</li> </ul>	<a href="#">Release notes</a>
7.5	<ul style="list-style-type: none"> <li>• Improved Kanban backlog: versions and epics panels to let you better manage your releases.</li> <li>• Sprint goals: add goals to your sprints to let your team know what you want to achieve.</li> <li>• Renaming and deleting inactive or closed sprints.</li> <li>• Jira Software Data Center on Microsoft Azure.</li> <li>• Events for creating and deleting issue links.</li> <li>• Syntax highlighting for additional 25 languages.</li> <li>• New Czech, Estonian, Danish, Icelandic, Norwegian, Romanian, Slovak, and Polish language packs.</li> </ul>	<a href="#">Release notes</a>

## Jira Service Management top features

Here's a summary of the great new features awaiting your users.

4.20	<p> <b>This is a Long Term Support release.</b></p> <ul style="list-style-type: none"> <li>• No new features to keep everything smooth and stable</li> </ul>	<a href="#">Release notes</a>
------	---	-------------------------------

4.19	<ul style="list-style-type: none"> <li>▪ Approvers from Insight asset management (Data Center)</li> <li>▪ Better language support for asset management</li> <li>▪ Manually adjusting projects to better change management (Data Center)</li> <li>▪ Data pipeline improvements (Data Center)</li> <li>▪ Pagination and UI improvements on Screens</li> </ul>	<a href="#">Release notes</a>
4.18	<ul style="list-style-type: none"> <li>▪ Another bunch of vulnerabilities fixed in the Jira platform and Jira Service Management</li> <li>▪ Showing SLAs on customer portals (Data Center)</li> <li>▪ Turning off search bar on customer portals (Data Center)</li> <li>▪ Mentions on customer portals (Data Center)</li> <li>▪ Summary of mobile app improvements</li> <li>▪ Analyzing how issues change over time with data pipeline</li> </ul>	<a href="#">Release notes</a>
4.17	<ul style="list-style-type: none"> <li>▪ Around 100 security vulnerabilities fixed in the Jira platform</li> <li>▪ Over 20 security vulnerabilities fixed in Jira Service Management</li> <li>▪ Custom fields now included in data pipeline export (Data Center)</li> </ul>	<a href="#">Release notes</a>
4.16	<ul style="list-style-type: none"> <li>▪ Get more insights into the usage of custom fields (Data Center)</li> <li>▪ Bulk delete custom fields (Data Center)</li> <li>▪ Default values for the Description field (Data Center)</li> <li>▪ Let users log in with multiple identity providers (Data Center)</li> <li>▪ Check the status of your data pipeline exports (Data Center)</li> <li>▪ Keep Jira safe by disabling basic authentication</li> </ul>	<a href="#">Release notes</a>
4.15	<ul style="list-style-type: none"> <li>▪ Mindville's Insight - Asset Management is now part of Jira Service Management (Data Center)</li> <li>▪ Customer portal improvements (cards view in Help Center, rich text editor, voting for requests)</li> <li>▪ Official support for service projects in the Jira mobile app</li> <li>▪ Improved data insights, now also for Jira Service Management</li> <li>▪ Displaying image attachments in email notifications</li> </ul>	<a href="#">Release notes</a>
4.14	<ul style="list-style-type: none"> <li>• Jira Service Desk is now Jira Service Management</li> <li>• New events in the audit log (Data Center)</li> <li>• Email templates made better (Data Center)</li> <li>• Personal access tokens</li> <li>• Embedded Crowd and password encryption</li> <li>• Accessibility improvements when creating issues</li> <li>• Improved data insights (Data Center)</li> </ul>	<a href="#">Release notes</a>
4.13	<p> <b>This is a Long Term Support release.</b></p> <ul style="list-style-type: none"> <li>▪ No new features to keep everything smooth and stable</li> </ul>	<a href="#">Release notes</a>

4.12	<ul style="list-style-type: none"> <li>• Managing Opsgenie incidents in Jira Service Management</li> <li>• Using Confluence Cloud as your knowledge base</li> <li>• OAuth 2.0 support for Microsoft</li> <li>• Changes to how users are displayed in the user picker</li> <li>• More control over your Advanced Audit log (Data Center)</li> <li>• Less app impact on indexing (Data Center)</li> <li>• Users created automatically with Just-in-time user provisioning (Data Center)</li> </ul>	<a href="#">Release notes</a>
4.11	<ul style="list-style-type: none"> <li>▪ Multilingual customer portal and help center</li> <li>▪ Advanced auditing (Data Center)</li> <li>▪ Managing private filters and dashboards</li> <li>▪ Embedded Crowd upgrade</li> <li>▪ Max timeout for the Favorite Filters gadget</li> <li>▪ Configuring how fast stale nodes are moved offline (Data Center)</li> </ul>	<a href="#">Release notes</a>
4.10	<ul style="list-style-type: none"> <li>▪ OAuth 2.0 support for incoming mail</li> <li>▪ User anonymization (GDPR) improvements</li> <li>▪ More insight into your custom fields (Data Center)</li> <li>▪ Stale nodes automatically removed (Data Center)</li> </ul>	<a href="#">Release notes</a>
4.9	<ul style="list-style-type: none"> <li>• Adding an option to control sharing settings for new requests created both via email and customer portal (addition to sharing settings from Jira Service Management 4.7)</li> <li>• Node and application status for Jira Data Center cluster monitoring</li> <li>• Accessibility: Text spacing</li> </ul>	<a href="#">Release notes</a>
4.8	<ul style="list-style-type: none"> <li>▪ Managing multiple issues at once with bulk actions</li> <li>▪ No more duplicate attachments in requests</li> <li>▪ Accessibility settings</li> <li>▪ Revamped audit log</li> <li>▪ Running Jira Data Center on a single node (available for all versions)</li> </ul>	<a href="#">Release notes</a>
4.7.	<ul style="list-style-type: none"> <li>▪ Requests are private by default</li> <li>▪ Email verification</li> <li>▪ Changes in how agents are treated by Jira</li> <li>▪ Anonymizing users for GDPR compliance</li> <li>▪ OpenID Connect (Data Center)</li> </ul>	<a href="#">Release notes</a>
4.6	<ul style="list-style-type: none"> <li>• New portal is now default</li> <li>• Improvements to agent queues</li> <li>• REST API for managing agent queues</li> <li>• Previous and next search results</li> <li>• Users and roles made better</li> <li>• Custom changes transferred on upgrade</li> <li>• Suffix search</li> <li>• New health check: JVM code cache</li> <li>• Rate limiting (Data Center)</li> <li>• New events in the audit log (Data Center)</li> </ul>	<a href="#">Release notes</a>

4.5	 <b>This is a Long Term Support release.</b> <ul style="list-style-type: none"> <li>▪ Massive performance improvements</li> </ul>	<a href="#">Release notes</a>
4.4	<ul style="list-style-type: none"> <li>• More control over SLA calendars</li> <li>• Issue archiving in Jira Service Management Data Center</li> <li>• Filters in 'Multi user picker' custom field</li> <li>• Batching emails as a default setting</li> <li>• External links open in a new tab</li> <li>• Filters to browse archived issues</li> <li>• Jira becomes more accessible</li> <li>• Lots of versions OK for boards</li> <li>• Time tracking in bulk edit</li> </ul>	<a href="#">Release notes</a>
4.3	<ul style="list-style-type: none"> <li>• Search just got smarter</li> <li>• Limited threads to boost performance</li> <li>• Content Delivery Network (CDN) for Jira Data Center</li> <li>• AdoptOpenJDK bundled with Jira</li> <li>• List of custom changes after upgrade</li> <li>• New filters to search for custom fields</li> <li>• Browsing projects is faster</li> <li>• Safer cluster lock mechanism for Data Center</li> <li>• Reindexing made easier</li> </ul>	<a href="#">Release notes</a>
4.2	<ul style="list-style-type: none"> <li>• Edit custom fields in Jira Service Management automation</li> <li>• Better knowledge base search experience</li> <li>• Enhanced knowledge base reports</li> <li>• Jira Service Management source code available</li> <li>• Support for Microsoft SQL Server 2017</li> <li>• Official support for Java 11</li> <li>• AdoptOpenJDK 8 bundled with Jira</li> </ul>	<a href="#">Release notes</a>
4.1	<ul style="list-style-type: none"> <li>• New help center and customer portal experience</li> <li>• Smarter automation webhooks</li> <li>• Information about recent activity on user profiles</li> <li>• New avatars</li> </ul>	<a href="#">Release notes</a>
4.0	<ul style="list-style-type: none"> <li>• Performance improvements for common tasks</li> <li>• New priority icons</li> <li>• New JQL options: finding authors of updates, and finding link types</li> <li>• New configuration for MySQL 5.7 that allows 4-byte characters</li> <li>• Massive improvements to indexing</li> </ul>	<a href="#">Release notes</a>
3.16	 <b>This is a Long Term Support release.</b> <ul style="list-style-type: none"> <li>• Support for OpenJDK 8</li> </ul>	<a href="#">Release notes</a>

3.15	<ul style="list-style-type: none"> <li>• Smart SLAs based on due dates</li> <li>• Custom fields optimizer (Data Center)</li> <li>• Sharing edit rights for filters and dashboards</li> <li>• New events in the audit log (filters and dashboards)</li> <li>• New look and feel of the custom fields page</li> </ul>	<a href="#">Release notes</a>
3.14	<ul style="list-style-type: none"> <li>• New look and feel for the most frequently used pages</li> <li>• Support for IPv6</li> <li>• Project archiving for Data Center</li> <li>• New columns on the Projects page (<i>Issues, Last issue update</i>)</li> <li>• Dutch added to languages available in Jira Service Management</li> </ul>	<a href="#">Release notes</a>
3.13	<ul style="list-style-type: none"> <li>• New events in the audit log</li> </ul>	<a href="#">Release notes</a>
3.12	<ul style="list-style-type: none"> <li>• Approve requests from email</li> <li>• Searching versions with a wildcard</li> <li>• IPv6 support for MySQL databases</li> </ul>	<a href="#">Release notes</a>
3.11	<ul style="list-style-type: none"> <li>• Improved quick search</li> </ul>	<a href="#">Release notes</a>
3.10	<ul style="list-style-type: none"> <li>• Visibility on approvals</li> <li>• Disable account verification emails</li> </ul>	<a href="#">Release notes</a>
3.9	 <b>This is a Long Term Support release.</b> <ul style="list-style-type: none"> <li>• Project-specific priorities (from 3.9.4 only)</li> <li>• Better canned responses</li> <li>• Auto approvals</li> <li>• Live monitoring of your Jira instance with JMX</li> </ul>	<a href="#">Release notes</a>
3.8	<ul style="list-style-type: none"> <li>• Import SLAs</li> <li>• Events for linked issues</li> <li>• Syntax highlighting for 25 more languages</li> </ul>	<a href="#">Release notes</a>
3.7	<ul style="list-style-type: none"> <li>• Canned responses comes to Jira Service Management</li> </ul>	<a href="#">Release notes</a>

## Supported platform changes

From time to time we will add or end support for a platform. Here's a summary of the changes since **Jira Software 7.4**, and **Jira Service Management 3.7**.

8.20 / 4.20	No changes	<a href="#">Supported platforms</a>
-------------	------------	-------------------------------------

<b>8.19</b> <b>/4.</b> <b>19</b>	<p><b>Added support for:</b></p> <ul style="list-style-type: none"> <li>▪ PostgreSQL 12</li> </ul> <p><b>End support for:</b></p> <ul style="list-style-type: none"> <li>• PostgreSQL 9.6</li> </ul>	Supported platforms
<b>8.18</b> <b>/4.</b> <b>18</b>	No changes	Supported platforms
<b>8.17</b> <b>/4.</b> <b>17</b>	<p><b>Added support for:</b></p> <ul style="list-style-type: none"> <li>• Microsoft SQL Server 2019</li> <li>• Microsoft Edge (Chromium)</li> </ul>	Supported platforms
<b>8.16</b> <b>/4.</b> <b>16</b>	No changes	Supported platforms
<b>8.15</b> <b>/4.</b> <b>15</b>	No changes	Supported platforms
<b>8.14</b> <b>/4.</b> <b>14</b>	No changes	Supported platforms
<b>8.13</b> <b>/4.</b> <b>13</b>	No changes	Supported platforms
<b>8.12</b> <b>/4.</b> <b>12</b>	<p><b>Ended support for:</b></p> <ul style="list-style-type: none"> <li>• MySQL 5.6</li> <li>• Microsoft SQL Server 2014</li> </ul> <p><b>Added support for:</b></p> <ul style="list-style-type: none"> <li>• MySQL 8.0</li> </ul>	Supported platforms
<b>8.11</b> <b>/4.</b> <b>11</b>	<p><b>Ended support for:</b></p> <ul style="list-style-type: none"> <li>▪ Hipchat app (also unbundled from Jira)</li> </ul>	Supported platforms
<b>8.10</b> <b>/4.</b> <b>10</b>	No changes	Supported platforms
<b>8.9</b> <b>/4.9</b>	No changes	Supported platforms

8.8 /4.8	<b>Ended support for:</b> <ul style="list-style-type: none"> <li>■ Microsoft SQL Server 2012</li> <li>■ PostgreSQL 9.4, 9.5</li> <li>■ Solaris</li> <li>■ Oracle 12c R1</li> </ul>	Supported platforms
8.7 /4.7	<b>Added support for:</b> <ul style="list-style-type: none"> <li>• PostgreSQL 11 and Aurora 3.0 for Jira Data Center</li> </ul>	Supported platforms
8.6 /4.6	<b>Added support for:</b> <ul style="list-style-type: none"> <li>• PostgreSQL 10</li> </ul> <b>Ended support for:</b> <ul style="list-style-type: none"> <li>• Internet Explorer 11</li> </ul>	Supported platforms
8.5 /4.5	No changes	Supported platforms
8.4 /4.4	<b>Added support for:</b> <ul style="list-style-type: none"> <li>• Aurora PostgreSQL for Jira Data Center</li> <li>• Oracle 19c</li> <li>• Oracle 18c</li> </ul> <b>Ended support for:</b> <ul style="list-style-type: none"> <li>• Deprecation of several Jira native importers available as part of the Jira Importers Plugin (JIM). We're only supporting import from CSV and JSON.</li> <li>• Advance notice: Createmeta REST endpoint will soon be removed.</li> </ul>	Supported platforms
8.3 /4.3	<b>Added support for:</b> <ul style="list-style-type: none"> <li>• Oracle 12c Release 2</li> </ul> <b>Ended support for:</b> <ul style="list-style-type: none"> <li>• Advanced notice: Ending support for several built-in importers in 8.4</li> <li>• Advanced notice: Removing Jira CDN dark feature in 8.4</li> </ul>	Supported platforms
8.2 /4.2	<b>Added support for:</b> <ul style="list-style-type: none"> <li>• Java 11</li> <li>• Microsoft SQL Server 2017</li> </ul> <b>Ended support for:</b> <ul style="list-style-type: none"> <li>• 32-bit installers</li> </ul>	Supported platforms
8.1 /4.1	No changes	Supported platforms

<b>8.0</b> <b>/4.0</b>	<p><b>Added support for:</b></p> <ul style="list-style-type: none"> <li>• Apache Tomcat 8.5.35 (replacing 8.5.32). Added in Jira 8.0.1.</li> </ul> <p><b>Ended support for:</b></p> <ul style="list-style-type: none"> <li>• PostgreSQL 9.3</li> <li>• MySQL 5.5</li> </ul>	Supported platforms
<b>7.13</b> <b>/3.</b> <b>16</b>	<p><b>Added support for:</b></p> <ul style="list-style-type: none"> <li>• OpenJDK 8</li> </ul>	Supported platforms
<b>7.12</b> <b>/3.</b> <b>15</b>	<p><b>Added support for:</b></p> <ul style="list-style-type: none"> <li>• Apache Tomcat 8.5.32 (replacing 8.5.29)</li> </ul>	Supported platforms
<b>7.11</b> <b>/3.</b> <b>14</b>	<p><b>Added support for:</b></p> <ul style="list-style-type: none"> <li>• Apache Tomcat 8.5.29 (replacing 8.5.6)</li> </ul>	Supported platforms
<b>7.10</b> <b>/3.</b> <b>13</b>	No changes.	Supported platforms
<b>7.9</b> <b>/3.</b> <b>12</b>	<p><b>Added support for:</b></p> <ul style="list-style-type: none"> <li>• Microsoft SQL Server 2016</li> </ul>	Supported platforms
<b>7.8</b> <b>/3.</b> <b>11</b>	No changes.	Supported platforms
<b>7.7</b> <b>/3.</b> <b>10</b>	No changes.	Supported platforms
<b>7.6</b> <b>/3.9</b>	No changes.	Supported platforms
<b>7.5</b> <b>/3.8</b>	No changes.	Supported platforms
<b>7.4</b> <b>/3.7</b>	<p><b>Added support for:</b></p> <ul style="list-style-type: none"> <li>• PostgreSQL 9.6</li> </ul> <p><b>Ended support for:</b></p> <ul style="list-style-type: none"> <li>• PostgreSQL 9.2</li> </ul>	Supported platforms

## Technical upgrade notes

This table is a high-level summary of the Jira Software and Jira Service Management upgrade notes. You should read the full upgrade notes before upgrading.

<b>8.20</b> /4. <b>20</b>	No important changes	<a href="#">Jira Software upgrade notes</a>  <a href="#">Jira Service Management upgrade notes</a>
<b>8.19</b> /4. <b>19</b>	<ul style="list-style-type: none"> <li>• Improved user assignment and mentioning performance</li> <li>• Jira Software binary installers bundled with Java 11</li> <li>• We've changed the data pipeline by adding disk space checks, export schema versioning, and improving exporting user details</li> <li>• We've added a configurable throttling mechanism to Activity Streams</li> </ul>	<a href="#">Jira Software upgrade notes</a>  <a href="#">Jira Service Management upgrade notes</a>
<b>8.18</b> /4. <b>18</b>	<ul style="list-style-type: none"> <li>▪ Bundled JRE disables secure connections to MySQL Community Edition 5.7.27 or older over TLS versions 1 and 1.1</li> <li>▪ We've upgraded the embedded H2 database</li> </ul>	<a href="#">Jira Software upgrade notes</a>  <a href="#">Jira Service Management upgrade notes</a>
<b>8.17</b> /4. <b>17</b>	<ul style="list-style-type: none"> <li>▪ User-generated custom fields and fields provided by apps are included in the data pipeline export by default</li> <li>▪ We've removed obsolete JIM importers</li> </ul>	<a href="#">Jira Software upgrade notes</a>  <a href="#">Jira Service Management upgrade notes</a>
<b>8.16</b> /4. <b>16</b>	<ul style="list-style-type: none"> <li>▪ Changes to data pipeline REST responses</li> </ul>	<a href="#">Jira Software upgrade notes</a>  <a href="#">Jira Service Management upgrade notes</a>

<b>8.15</b> <b>/4.</b> <b>15</b>	<ul style="list-style-type: none"> <li>■ We've made small tweaks to some of the email templates. These changes are optional for you.</li> <li>■ We've made some changes to the web.xml file, so make sure to not copy over the old file after upgrading. This could affect attachments in Jira.</li> <li>■ Jira Software: If you're an existing user of Mindville's Insight - Asset Management app, there is some important info that might affect you. Insight has been bundled with Jira Service Management, so make sure to read more about it in the upgrade notes.</li> </ul>	<a href="#">Jira Software upgrade notes</a>  <a href="#">Jira Service Management upgrade notes</a>
<b>8.14</b> <b>/4.</b> <b>14</b>	<ul style="list-style-type: none"> <li>■ For Data Center, we'll move email templates from Jira resources to your Jira shared home directory on upgrade. If you customized the templates, read the upgrade notes for guidance on how to reapply your changes in Jira 8.14.</li> </ul>	<a href="#">Jira Software upgrade notes</a>  <a href="#">Jira Service Management upgrade notes</a>
<b>8.13</b> <b>/4.</b> <b>13</b>	<p>No important changes</p>	<a href="#">Jira Software upgrade notes</a>  <a href="#">Jira Service Desk upgrade notes</a>
<b>8.12</b> <b>/4.</b> <b>12</b>	<p>Important notes:</p> <ul style="list-style-type: none"> <li>• Known issue: Azure SQL is reported as unsupported</li> <li>• Known issue: MySQL 8.0 is reported as misconfigured</li> <li>• We've added database password encryption.</li> <li>• We've extended API response to look for archived projects.</li> <li>• We've added new events in the Advanced audit log for Data Center.</li> <li>• We fixed a bug that caused issues removed from sprint not showing in Burndown Chart or Sprint Report. This required full reindex.</li> </ul>	<a href="#">Jira Software upgrade notes</a>  <a href="#">Jira Service Desk upgrade notes</a>
<b>8.11</b> <b>/4.</b> <b>11</b>	<p>Important notes:</p> <ul style="list-style-type: none"> <li>• We fixed a bug that caused issues removed from sprint not showing in Burndown Chart or Sprint Report. This requires full re-index.</li> <li>• We fixed fetching updates from different products in Jira.</li> <li>• We upgraded Embedded Crowd in Jira from version 2.0 to 4.0.</li> <li>• We upgraded Apache Tomcat from 8.5.50 to 8.5.56.</li> <li>• Hipchat got unbundled from Jira.</li> </ul>	<a href="#">Jira software upgrade notes</a>  <a href="#">Jira Service Desk upgrade notes</a>

<b>8.10</b> <b>/4.10</b>	<p>Important notes:</p> <ul style="list-style-type: none"> <li>• If you're using Google or Microsoft for incoming mail, we recommend that you create an OAuth 2.0 integration, and reconfigure your mail servers and email channels.</li> </ul>	<a href="#">Jira software upgrade notes</a>  <a href="#">Jira Service Desk upgrade notes</a>
<b>8.9</b> <b>/4.9</b>	<p>Important notes:</p> <ul style="list-style-type: none"> <li>• New API to help the Versions view load faster</li> </ul>	<a href="#">Jira software upgrade notes</a>  <a href="#">Jira Service Desk upgrade notes</a>
<b>8.8</b> <b>/4.8</b>	<p>Important notes:</p> <ul style="list-style-type: none"> <li>▪ The new audit log requires that we migrate your current audit log entries. This might take some time, but it will be done in the background.</li> <li>▪ We've enabled Garbage First Garbage Collection (G1 GC) for Jira instances running on Java 11.</li> </ul>	<a href="#">Jira software upgrade notes</a>  <a href="#">Jira Service Desk upgrade notes</a>
<b>8.7</b> <b>/4.7</b>	<p>Important notes:</p> <ul style="list-style-type: none"> <li>• PostgreSQL 11 support</li> <li>• OpenID Connect support for Jira Data Center</li> <li>• Changes in the issue collector</li> <li>• Disabling HTML in custom fields description</li> <li>• Changing how agents are treated in SLAs and automation rules</li> <li>• Changed default behavior of sharing new requests with the customers organization</li> </ul>	<a href="#">Jira Software upgrade notes</a>  <a href="#">Jira Service Desk upgrade notes</a>
<b>8.6</b> <b>/4.6</b>	<p>Important notes:</p> <ul style="list-style-type: none"> <li>• API change for JXM monitoring</li> <li>• Support for PostgreSQL 10</li> <li>• New JVM code cache check</li> <li>• New information in the audit log</li> <li>• Transferring custom modifications from oldJira files</li> <li>• Better support for reply emails from Outlook by mailhandlers</li> </ul>	<a href="#">Jira Software upgrade notes</a>  <a href="#">Jira Service Desk upgrade notes</a>

<p><b>8.5</b> <b>/4.5</b></p>	<p>Important notes:</p> <ul style="list-style-type: none"> <li>• This is the last release to support IE 11</li> </ul>	<p><a href="#">Jira Software upgrade notes</a></p> <p><a href="#">Jira Service Desk upgrade notes</a></p>
<p><b>8.4</b> <b>/4.4</b></p>	<p>Important notes:</p> <ul style="list-style-type: none"> <li>• To maximize your data security we've made the Public Sharing feature OFF by default. We also recommend that you revise your filter, dashboards and user permissions.</li> <li>• We've introduced extra security measures to make issue attachments and keys safe and inaccessible for anonymous users.</li> <li>• To help Outlook users respond to Jira notifications we've improved the <i>RegexCommentHandler</i> UI to make it easier to set it up properly. We also advise that you don't use the <i>NonQuotedCommentHandler</i> as it doesn't work with html or rich text emails.</li> <li>• Email notifications will be batched by default. You can still disable the feature and the feature will remain disabled if you disabled it in a previous version of Jira.</li> <li>• We've introduced a new way of generating user keys for new users to reduce the number of places we store personal data. All user keys for new users (no changes for existing users) have the following format: <code>JIRAUSER10100</code>.</li> <li>• If you choose to use one of the new supported databases, Oracle 18c or 19c, you need to download a new JDBC driver. See upgrade notes for details.</li> <li>• We're upgrading Tomcat to version 8.5.42.</li> </ul>	<p><a href="#">Jira Software upgrade notes</a></p> <p><a href="#">Jira Service Desk upgrade notes</a></p>
<p><b>8.3</b> <b>/4.3</b></p>	<p>Important notes:</p> <ul style="list-style-type: none"> <li>• Jira is now bundled with AdoptOpenJDK 8 instead of Oracle's JDK. If you're installing Jira manually, you need to install JDK.</li> <li>• We have upgraded the atlassian-jslibs plugin from version 1.2.5 to 1.4.1 and are now bundling React 16.8.6, ReactDOM 16.8.6 factory, and Marionette 4.1.2 with Jira.</li> <li>• When you upgrade you need to reapply any custom changes you've made to your old Jira files to the same files in the upgraded version. We will show you a list of files with custom changes.</li> <li>• To fix issues importing a backup from Jira Cloud into Jira Server, we needed to change the database column size. Now, Jira might start up slower after upgrade, especially if you're using MySQL database.</li> <li>• You can now encrypt your database password stored in the <code>dbconfig.xml</code> file.</li> <li>• We improved the cluster lock mechanism in Jira DC by introducing lock timeouts. When a node loses a cluster lock it will be automatically recoverable without any actions from administrator.</li> </ul>	<p><a href="#">Jira Software upgrade notes</a></p> <p><a href="#">Jira Service Desk upgrade notes</a></p>

<p><b>8.2</b> <b>/4.1</b></p>	<p>Important notes:</p> <ul style="list-style-type: none"> <li>• We've upgraded the Microsoft JDBC driver, bundled with Jira, to 7.2.1. This version is required for Java 11 support.</li> <li>• Jira installers are now bundled with AdoptOpenJDK JRE 8 instead of Oracle JDK. If you're using the installer, check your Java version after the upgrade, as Jira might have switched to AdoptOpenJDK.</li> </ul> <p>We've made some look and feel changes to the help center and customer portal. This is the first round of improvements aimed at making the customer experience both simple and polished.</p> <p>In order to do this, we've had to make some changes to the frontend codebase and this may impact third-party apps.</p> <p><b>What's the potential impact?</b></p> <p>There's two things to be aware of:</p> <ul style="list-style-type: none"> <li>• layout changes may impact how your app looks within the customer portal</li> <li>• the customer portal now loads routes asynchronously, which can impact previously available globals.</li> </ul> <p><b>What you need to do</b></p> <p>In 4.2, your app might be impacted by these changes if no action is taken.</p> <p>Steps to take:</p> <ol style="list-style-type: none"> <li>1. Check that the dependencies your app needs are specified in your web resources. When checking this, consider if these dependencies are necessary, or if they're negatively impacting load-time performance. You can use <a href="https://bitbucket.org/atlassianlabs/atlassian-webresource-webpack-plugin">https://bitbucket.org/atlassianlabs/atlassian-webresource-webpack-plugin</a> to help achieve this.</li> <li>2. Check that your app works 'as expected' within the new layout. When the Service Management admin activates the new layout, the &lt;body&gt; element will contain the class custom-portal-layout-flag.</li> </ol>	<p><a href="#">Jira Software upgrade notes</a></p> <p><a href="#">Jira Service Desk upgrade notes</a></p>
-----------------------------------	---	---

<p><b>8.1</b> <b>/4.1</b></p>	<p><b>Important notes:</b></p> <ul style="list-style-type: none"> <li>• We are deprecating 32-bit installers, and they will be removed in the upcoming release.</li> <li>• We are deprecating the CDN dark feature, and are planning to remove it in the upcoming release. We'll replace it with an official CDN support for Jira Data Center.</li> </ul> <p>We've made some look and feel changes to the help center and customer portal. This is the first round of improvements aimed at making the customer experience both simple and polished.</p> <p>In order to do this, we've had to make some changes to the frontend codebase and this may impact third-party apps.</p> <p><b>What's the potential impact?</b></p> <p>There's two things to be aware of:</p> <ul style="list-style-type: none"> <li>• layout changes may impact how your app looks within the customer portal</li> <li>• the customer portal now loads routes asynchronously, which can impact previously available globals.</li> </ul> <p><b>What you need to do</b></p> <p>In Jira Service Desk 4.1, your app should work as expected, but you'll need to check your dependencies and take the recommended action outlined below. In 4.2, your app might be impacted by these changes if no action is taken.</p> <p>Steps to take:</p> <ol style="list-style-type: none"> <li>1. Check that the dependencies your app needs are specified in your web resources. When checking this, consider if these dependencies are necessary, or if they're negatively impacting load-time performance. You can use <a href="https://bitbucket.org/atlassianlabs/atlassian-webresource-webpack-plugin">https://bitbucket.org/atlassianlabs/atlassian-webresource-webpack-plugin</a> to help achieve this.</li> <li>2. Check that your app works 'as expected' within the new layout. When the Service Management admin activates the new layout, the &lt;body&gt; element will contain the class custom-portal-layout-flag.</li> </ol>	<p><a href="#">Jira Software upgrade notes</a></p> <p><a href="#">Jira Service Desk upgrade notes</a></p>
-----------------------------------	--	---

<p><b>8.0</b> <b>/4.0</b></p>	<p><b>Important notes:</b></p> <ul style="list-style-type: none"> <li>• <b>Incompatible index:</b> Due to the upgrade of the library responsible for the Jira index, your current index will become incompatible. You'll need to create a new index by reindexing Jira after the upgrade.</li> <li>• <b>Index location changes:</b> Starting from Jira 8.0, the Jira index will be stored in a new location: <code>&lt;Jira-home-directory&gt;/caches/indexesV1</code>.</li> <li>• <b>Automatic reindex:</b> Because of the above changes, Jira will trigger an automatic reindex after the upgrade, which might mean you'll need to reindex twice following the upgrade. Since this might be really time-consuming for large Jira instances, you can disable the automatic reindex, and run it manually whenever you're ready.</li> <li>• <b>Incompatible apps:</b> You'll need to disable all apps that are incompatible with the new version before the upgrade. Incompatible apps might block the upgrade or the Jira startup after the upgrade.</li> <li>• <b>Reduced logging:</b> Application log output ( <code>atlassian-jira.log</code> ) is no longer mirrored in the Tomcat log file, <code>catalina.out</code> , to preserve the disk space.</li> <li>• <b>Jira needs more memory:</b> We've increased the default heap size available to Jira to 2GB. Note that this change requires extra steps if you're using 32-bit systems.</li> <li>• <b>Zero downtime upgrades:</b> This upgrade method won't be available when upgrading from Jira 7.x to Jira 8.x. Once you're on Jira 8.x, you can use it to upgrade to any later version.</li> </ul> <p>In Jira Service Desk 4.0, we've removed <code>com.atlassian.fugue</code>, and updated our APIs to use Core Java Data types and Exceptions. We've introduced this change to make it easier to develop on Jira Service Management.</p> <p>What you'll need to do</p> <p>Before using Core Java Data types and Exceptions, update any scripts, integrations, or apps that make requests to endpoints returning <code>com.atlassian.fugue</code>. This stops them breaking after the update.</p> <p>Read our <a href="#">Java API docs</a> and <a href="#">REST API docs</a> to learn how.</p>	<p><a href="#">Jira Software upgrade notes</a></p> <p><a href="#">Jira Service Desk upgrade notes</a></p>
<p><b>7.13</b> <b>/3.</b> <b>16</b></p>	<p><b>Important notes:</b></p> <ul style="list-style-type: none"> <li>• After upgrading Apache Tomcat to 8.5.32, we've added new properties to the <code>server.xml</code> file. Make sure you don't copy the old files to the new Jira version, but add the missing properties (or copy your customizations to the new file).</li> <li>• New properties, controlling the JVM code cache, have been added to the <code>setenv.sh</code> / <code>.bat</code> file.</li> <li>• Jira Service Desk 3.16 has deprecated the use of <code>com.atlassian.fugue</code>. In Jira Service Desk 4.0, we'll be permanently removing it and updating our APIs to use Core Java Data types and Exceptions instead.</li> </ul>	<p><a href="#">Jira Software upgrade notes</a></p> <p><a href="#">Jira Service Desk upgrade notes</a></p>

<p><b>7.12</b> <b>/3.</b> <b>15</b></p>	<p><b>Important notes:</b></p> <ul style="list-style-type: none"> <li>• Jira Service Desk 3.15 has deprecated the use of <code>com.atlassian.fugue</code>. In Jira Service Desk 4.0, we'll be permanently removing it and updating our APIs to use Core Java Data types and Exceptions instead.</li> </ul>	<p><a href="#">Jira Software upgrade notes</a></p> <p><a href="#">Jira Service Desk upgrade notes</a></p>
<p><b>7.11</b> <b>/3.</b> <b>14</b></p>	<p><b>Important notes:</b></p> <ul style="list-style-type: none"> <li>• We've upgraded Apache Tomcat to 8.5.29. There might be some changes to Tomcat config files, so make sure you don't copy the old files to the new Jira version. See upgrade notes for details.</li> <li>• We support IPv6, but recommend that you use the mixed mode (IPv4 + IPv6). If your systems are IPv6 only, with IPv4 disabled, some Jira features won't work.</li> </ul>	<p><a href="#">Jira Software upgrade notes</a></p> <p><a href="#">Jira Service Desk upgrade notes</a></p>
<p><b>7.10</b> <b>/3.</b> <b>13</b></p>	<p>No important changes.</p>	<p><a href="#">Jira Software upgrade notes</a></p> <p><a href="#">Jira Service Desk upgrade notes</a></p>
<p><b>7.9</b> <b>/3.</b> <b>12</b></p>	<p><b>Important notes:</b></p> <ul style="list-style-type: none"> <li>• Jira Data Center changes cache replication to asynchronous. No immediate actions are required, but make sure you read about this change, and check whether you need to provide extra disk space in the local home directory on each node.</li> </ul>	<p><a href="#">Jira Software upgrade notes</a></p> <p><a href="#">Jira Service Desk upgrade notes</a></p>
<p><b>7.8</b> <b>/3.</b> <b>11</b></p>	<p><b>Important notes:</b></p> <ul style="list-style-type: none"> <li>• Oracle JDBC driver is no longer shipped with Jira. You'll need to download it from the Oracle website, and copy to your Jira installation directory (<code>/lib</code>).</li> </ul>	<p><a href="#">Jira Software upgrade notes</a></p> <p><a href="#">Jira Service Desk upgrade notes</a></p>

<p><b>7.7</b> <b>/3.10</b></p>	<p><b>Important notes:</b></p> <ul style="list-style-type: none"> <li>• Oracle JDBC driver is no longer shipped with Jira. You'll need to download it from the Oracle website, and copy to your Jira installation directory (<code>/lib</code>).</li> <li>• Account verification emails are sent out to Jira Service Desk customers when requests are raised by email. After the upgrade, project administrators can disable these account verification emails.</li> </ul> <p><b>Known issues:</b></p> <ul style="list-style-type: none"> <li>• A fix for duplicated job IDs requires extra steps in some instances, just to be sure (checking the database if the index was created; restarting your Jira instance.)</li> </ul>	<p><a href="#">Jira Software upgrade notes</a></p> <p><a href="#">Jira Service Desk upgrade notes</a></p>
<p><b>7.6</b> <b>/3.9</b></p>	<p><b>Important notes:</b></p> <ul style="list-style-type: none"> <li>• Priority schemes (UI changes). We've made significant changes to how priorities are managed in Jira, but these changes are only available from Jira Service Desk 3.9.4.</li> <li>• We're no longer shipping the Oracle JDBC driver with Jira. You'll need to download it from the Oracle site and move to the <code>/lib</code> directory in the Jira installation directory.</li> </ul> <p><b>Known issues:</b></p> <ul style="list-style-type: none"> <li>• A fix for duplicated job IDs requires extra steps in some instances, just to be sure (checking the database if the index was created; restarting your Jira instance.)</li> </ul>	<p><a href="#">Jira Software upgrade notes</a></p> <p><a href="#">Jira Service Desk upgrade notes</a></p>
<p><b>7.5</b> <b>/3.8</b></p>	<p><b>Important notes:</b></p> <ul style="list-style-type: none"> <li>• New JDBC driver for Microsoft SQL Server requires a new URL. If you didn't modify the original URL, you can fix it with Jira configuration tool. If you customized the URL, you might need to change it manually.</li> </ul>	<p><a href="#">Jira Software upgrade notes</a></p> <p><a href="#">Jira Service Desk upgrade notes</a></p>
<p><b>7.4</b> <b>/3.7</b></p>	<p><b>Important notes:</b></p> <ul style="list-style-type: none"> <li>• New options added to the startup files to enable garbage collection. Just a heads up, no actions required.</li> </ul>	<p><a href="#">Jira Software upgrade notes</a></p> <p><a href="#">Jira Service Desk 3.7 upgrade notes</a></p>



All ready?

Head back to the landing page, and choose your upgrade method or see our nifty upgrade checklist.

[Go to landing page](#)

# Upgrade methods

Choosing the upgrade method depends on the specifics of your environment. Below you can find information that will help you choose the right one for you. If you don't have any particular requirements and just want to upgrade Jira, pick one of the regular methods.

- [Regular methods](#)
- [Additional options](#)
- [Migrating Jira](#)

## Regular methods

These are the regular methods of upgrading Jira. Make sure you also check additional options below them.

### Installer

#### Server | Data Center

This method includes the use of a typical installation wizard that will guide you through all the stages of upgrading Jira. On each of the panels, you'll need to provide necessary information, such as your user credentials, or details needed to connect to the database.

**What you'll download:** .bin or .exe file

[Go for Server or Data Center \(non-clustered\)](#)

[Go for Data Center \(clustered\)](#)

### Manual

#### Server | Data Center

This method includes downloading Jira files compressed into a .zip or .tar.gz package, and extracting them into the chosen location. It doesn't include a typical installation (as you'll extract an almost-ready Jira instance), and so is much faster. You'll provide all details needed to connect Jira to your database by manually editing the right files.

**What you'll download:** .tar.gz or .zip archive

[Go for Server or Data Center \(non-clustered\)](#)

[Go for Data Center \(clustered\)](#)

## Additional options

These options reuse the installer or manual, but contain extra pre- and post-upgrade steps.

## Zero downtime

### Data Center

Upgrade with zero downtime is a special method available for Jira Data Center. It introduces the *upgrade mode* that allows your nodes to work on different Jira versions while you upgrade them one by one. During the upgrade, Jira remains fully functional and open to your users. Its performed by using either the installer or manual - its a classical upgrade, just with extra pre-upgrade and post-upgrade steps.

**What you'll download:** installer or manual (see [above](#))

[Go for Data Center](#)

#### Unavailable for Jira 7.x to Jira 8.x upgrades

This method is not available when upgrading from Jira 7.x to Jira 8.x. Once you're on Jira 8.x, you can use it to upgrade to any later version.

## Fallback

### Server | Data Center

This method allows you to safely roll back to your previous Jira version if the upgrade takes longer than expected, or if you encounter any issues. It requires that you set up a proxy server to have a quick way of redirecting your users either to the existing or to the new instance of Jira, depending on whether you're happy with the upgrade or not. Its a good fit for environments where Jira is mission-critical, and you cant allow prolonged downtime.

**What you'll download:** installer or manual (see [above](#))

[Go for Server or Data Center \(non-clustered\)](#)

[Go for Data Center \(clustered\)](#)

## Migrating Jira

Use the following methods to migrate your Jira instance to a different server, or to migrate between Jira Cloud and Jira Server.

### Migrating Jira to another server

#### Server | Data Center

This method lets you migrate your Jira applications to a different server, which includes new operating system, new locations for storing your index or attachments, or new database or database system. To migrate Jira, youll need to install a new Jira instance, and then migrate your existing data between the databases, finally also copying your home directory and any existing customizations.

**What you'll download:** installer or manual (see [above](#))

[Go for Server and Data Center](#)

## Migrating from Jira Cloud to Jira Server

### Server | Data Center

This method is for people who are currently using Jira Cloud, and wish to move to Jira Server (a Jira installation hosted on your own servers). Jira Cloud might be ahead of Jira Server, or focusing on slightly different areas, which means that some features may not be available after youve moved to Jira Server. The migration will include installing Jira Server from scratch, and then moving your data from Jira Cloud to your new database.

**What you'll download:** installer or manual (see [above](#))

[Go for Server and Data Center](#)

## Migrating from Jira Server to Jira Cloud

### Server | Data Center

This method is for people who are using Jira Server and wish to move to Jira Cloud. Currently, you can migrate in two ways:

- Use the Jira import site (Server and Data Center)
- Use the Cloud Migration Assistant that allows you to migrate your projects, users, and groups (Server)

[Go for Server and Data Center](#)

# Creating a test environment for Jira

When you upgrade Jira, we strongly recommend performing the upgrade in a test environment before upgrading your production site.

The suggested license to apply to a non-production environment is a [Developer License](#), not an Evaluation license. You should only use an Evaluation license if you don't have a Developer license tied to your main license.

- [Get a free license to use in testing environment](#)
- [Replicate database](#)
- [\(DC\) Remove production nodes from the `clusternodetable`](#)
- [Update production URL on text gadgets](#)
- [Delete mail server data](#)
- [Remove applinks](#)
- [Replicate Jira](#)
- [\(DC\) Manage shared home directory](#)
- [Start Jira](#)
- [Update webhooks](#)
- [\(DC\) Additional steps](#)
- [Change the look and feel](#)

## Replicate your environment

Your test environment should replicate your real-live environment (production), including any reverse proxies, SSL configuration, or load balancer (for Data Center). You can decide to use a different physical server or a virtualized solution but make sure it is an appropriate replica of your production environment.

For the purposes of these instructions, we assume your test environment is physically separate from your production environment, and has the same operating system (and Java version if you've installed Jira manually).

## Create a test environment

## 1. Replicate your database

To replicate your database:

1. Back up your production database. Refer to the documentation for your database for more info on the best way to do this.
2. Install your database on the test server and restore the backup.

The steps for restoring your database backup will differ depending on your chosen database and backup tool. Make sure:

- Your new test database has a **different** name from your production database.
- Your test database user account has the **same** username and password as your production database user account.
- Character encoding and other configurations are the same as your production database (for example character encoding should be Unicode UTF-8 (or AL32UTF8 for Oracle databases).

**i** If you're using the PostgreSQL database, be aware that replication may cause fragmentation of the data and indexes. In its turn, the fragmentation may slow down your environment, especially during the reindex operation.

To improve the environment's performance, after replicating the database, you should run the following database maintenance commands or tasks.

1. **VACUUM** reclaim the storage space used by stale data. You may also use the **ANALYZE** command alongside. So, a new query plan will be created for Jira's database. The queries will be optimized to use the most appropriate indexes.
2. **REINDEX** rebuild the index using the data from the indexes table. So, the old copy of the index will be replaced.

Learn more about the commands in the document [Optimize and Improve PostgreSQL Performance with VACUUM, ANALYZE, and REINDEX](#).

### CLUSTER

## 2. Remove production/abandoned nodes from the `clusternode` table

This is to make sure the production and test nodes do not behave like one cluster. To do that:

1. Review the content of the `clusternode` table. This table is the reference used by Jira when it comes to Data Center operations. Execute the following command:

```
SELECT * FROM clusternode;
```

2. Remove any production node records.
3. Remove any abandoned node records as they might affect reindexing operations later on. To learn more on how to do that, see [this](#) article.

**i** backup also brings abandoned nodes to the `clusternode` table.

Additionally, make sure the production and the test environment are in a different subnet. This is to prevent the test environment to try and communicate with production. If clusters see each other within network, it might cause issues.

### 3. Update the production URL on text gadgets

Text gadgets allow custom html text to be displayed on dashboards.

If you do not update the URL, the dashboard links on the test environment will redirect to the production environment, and users will get a dead page as a result.

Replace the old URL with the new one. For example, use the following command for PostgreSQL:

```
update gadgetuserpreference set userprefvalue = REPLACE(userprefvalue,'//prod.jira.base.url/', '//test.jira.base.url/') where userprefvalue like '%//prod.jira.base.url/%';
```

For MSSQL:

```
update gadgetuserpreference set userprefvalue = cast(replace(cast(userprefvalue as nvarchar(max)), '//prod.jira.base.url/', '//test.jira.base.url/') as next) where userprefvalue like '%//prod.jira.base.url/%';
```

Remember to replace the production and test.jira.base.url in the command with your production and test Jira base URLs. For example, [jira.atlassian.com](https://jira.atlassian.com) and [jira-test.atlassian.com](https://jira-test.atlassian.com).

### 4. Delete mail server data

To prevent emails being sent from the test environment and issues being created in this environment rather than in production, delete all incoming and outgoing mail servers by truncating the `mailserver` table.

Additionally, remove all mail handlers by running the following command:

```
delete from serviceconfig where clazz='com.atlassian.jira.service.services.mail.MailFetcherService'
```

To make sure no email is sent from the test environment, it's also a good idea to:

- Remove all subscriptions in Jira database with:

```
delete from filtersubscription;
```

- Remove notification schemes from all projects in Jira database with:

```
delete from nodeassociation where sink_node_entity='NotificationScheme'
```

### 5. (Optional) Remove applinks

It's a good practice to remove any application links between Jira and other Atlassian applications so that the links in production and test environment don't get confused. Alternatively, you can keep the links and just update them (see step 11) however removing them altogether is less problematic. For details, see [Remove an application link from Jira server using SQL](#).

## 6. Replicate Jira

To replicate Jira, make a copy of your Jira installation and point it to your test database.

1. Copy your entire **production installation directory** to your test server.
2. Copy your entire **production home directory** to your test server.
3. Edit `<installation-directory>/atlassian-jira/WEB-INF/classes/jira-application.properties` to point to your test home directory.  
CLUSTER For DC, make this change on every node.
4. Edit `<home-directory>/dbconfig.xml` or `<installation-directory>/server.xml` (older versions) to point to your test database.

 Make sure your test environment is not pointing to your production database.

CLUSTER

## 7. Manage shared home directory

1. Copy the **production shared home directory** to the test server.
2. Edit `<local-home-directory>/cluster.properties` to point to your test shared home directory. Make this change on every test node.

## 8. Start Jira in test environment

1. Start Jira with the following **System Properties** to make sure your test site does not send or receive notifications and emails. For more info about disabling email, see [Disable email sending/receiving](#).

```
-Datlassian.notifications.disabled=true
-Datlassian.mail.senddisabled=true
-Datlassian.mail.fetchdisabled=true
-Datlassian.mail.popdisabled=true
```

CLUSTER Start one node at a time.

If that's the case, you can keep email notifications enabled and [prepare a development server's mail configuration](#).

2. Head to `http://localhost:<port>` and log in to Jira on your test server.
3. Go to **Administration**  **> System > General Configuration**, and change the **base URL** of your test site (for example `mysite.test.com`).
4. Go to **Administration**  **> Applications > Versions and licenses**, and apply your development license. To update the license, click the edit icon next to it.
5. Go to **Administration**  **> System > System info**, and check that Jira is correctly pointing to your test database, and test home directory.
6. Go to **Administration**  **> System > Look and feel**, and change the colors of the test instance to make it different from the production instance. That's a small change, but it might help you avoid big mistakes.

## 8. Update webhooks

Your test environment shouldn't be able to trigger production webhooks. This is why you need to edit them in the new environment and either disable them or change to test webhooks. See how to [Manage webhooks](#).

## CLUSTER

**10. (Optional) Replicate external user management**

If you're managing users in Crowd or an external directory you can:

- replicate Crowd or your external directory in your test environment and point your Jira test site to your test external directory (recommended).
- provide your test server with network or local access to the same hosts as your production server.

**11. Modify application links**

If you have application links between Jira and other Atlassian applications and you have not deleted them (see step 5), you should change the server ID on each test application. See [How to change the server ID of Confluence](#) and [Changing Server ID for Test Installations](#) for Jira.

If you don't change the server ID and update your application links there is a chance that when you create a new application link in production it will point to your test server instead.

**12. (Optional) Change the look and feel**

It's a good practice to change the look and feel of the test instance to have a different logo or color scheme than the production environment. This will help users to distinguish between the two environments and not rely solely on the URL.

See [Configuring the look and feel of your Jira applications](#).



All good?

Head back to the landing page, and complete the pre-upgrade steps.

[Go to landing page](#)

# Preparing for the upgrade

All upgrade methods share common pre-upgrade steps that help you prepare for the upgrade. At this point, you shouldn't have to make any changes to your production Jira.

- [Review upgrade notes](#)
- [Run health check](#)
- [Check compatibility of apps](#)
- [Back up your Jira instance](#)

## Review upgrade notes

With every Jira release, we publish upgrade notes that might include known issues, important notes related to the upgrade, or additional steps that you need to complete. We've summed up all upgrade notes in our upgrade matrix, so you can quickly check whether you need to make some additional changes for your upgrade.

- [View the summary of upgrade notes](#)

## Run health check

Jira has several health checks that let you verify whether your Jira instance is ready for an upgrade.

1. Go to **Administration** (⚙️) > **System** > **Troubleshooting and support tools**.

Check the results of all instance health checks. Every health check will have a brief description of what it does, and a link to an article with more information, so you know how to fix any detected problems.

2. Make sure that checks related to the license and database don't report any problems, but you can also fix other detected problems.

The screenshot shows the 'Troubleshooting and support tools' page in Jira. At the top, there's a header with the title and a 'Give feedback' link. Below the header, there's a sub-header 'Instance health checks' with a brief description. A list of checks is shown, all with green checkmarks indicating they passed. The checks are: Supported Platforms, File system, Database, File system, Indexing, Attachments, Cluster, and Secondary Storage. At the bottom of the list, it says 'Last Run: 18 minutes ago'. On the right side, there's a 'Notifications' box with text explaining that the system will check the health of the instance periodically and can notify when there's a problem. It also mentions that if you're seeing too many notifications, you can limit them below, with a 'Show all notifications' button.

## Check compatibility of apps (add-ons)

Make sure your apps are compatible with the new version, so they keep working after the upgrade. You'll need to disable all incompatible apps as they might affect the upgrade.

1. Go to **Administration** (⚙️) > **Manage apps**.

2. Scroll down past the list of apps from Atlassian Marketplace, and click **JIRA update check**.
3. Select a version you want to upgrade to, and click **Check**.

### JIRA update check

The Universal Plugin Manager can help you prepare for a JIRA update. Choose one of the newer versions below and click **Check**. This page examines and reports on the compatibility of the add-ons you installed.

Check compatibility for update to:

#### Compatible

These add-ons are compatible with JIRA 8.0. No action needed.

>  Adaptavist ScriptRunner for JIRA	<input type="button" value="Disable"/>
>  Agile Cards	<input type="button" value="Disable"/>
>  Chat for Service Desk	<input type="button" value="Enable"/>
>  Portfolio for Jira	<input type="button" value="Disable"/>

#### Unknown

4. Depending on the compatibility results for your apps, **apply the actions** described below.

Status	Details
<b>Incompatible</b>	Your apps are incompatible with the new version. <b>Action:</b> Disable all incompatible apps before you proceed with the upgrade.
<b>Compatible</b>	Your apps are compatible with the new version. <b>Action:</b> No action needed, you're good to go.
<b>Compatible, if updated</b>	Your apps will be compatible with the new version once you upgrade them. <b>Action:</b> Upgrade your apps before you proceed with the upgrade. When you run the <i>Jira update check</i> again, they will be marked as <b>compatible</b> .
<b>Compatible once both are updated</b>	Your apps will be compatible with the new version once you upgrade both Jira and these apps. <b>Action:</b> We recommend that you disable these apps and proceed with the Jira upgrade. Once you're on a new version, you can upgrade the apps and enable them back. <ol style="list-style-type: none"> <li>1. Disable these apps, because they're incompatible at this point.</li> <li>2. Upgrade Jira to a new version.</li> <li>3. Upgrade the apps.</li> <li>4. Enable the apps.</li> </ol>
<b>Unknown</b>	We can't check the compatibility of this app. It usually applies to custom apps. <b>Action:</b> It's safer to disable this app, and check how it behaves in a testing environment.

### Get your list of files with custom modifications

Have you modified files to customize your Jira? If you want to keep these changes, make sure you know which files have been modified.

- If you're on the latest version of the [ASTS plugin](#), go to **Administration** (⚙️) > **Applications** > **Plan your upgrade** to see the list of files in which you introduced custom changes. If you want to keep these changes in your upgraded instance, you need to copy the changes (not entire files!) to the respective files during upgrade. Have this list handy, it will save you time during the upgrade process.
- If you cannot pull up the **Plan your upgrade** page, compile the list of files you've modified.

For more, see [Pre-upgrade planning tool](#).

**i** If you missed any of the changed files, worry not. We'll show you the files that contain changes that have not been copied over when you start your Jira after upgrade.

## Back up your Jira instance

Back up the Jira database and important directories, so you can safely roll back to your previous setup if something goes wrong.

### Database

Use the database native tools to create the backup. If your database doesn't support online backups, you'll need to stop Jira first.

### Jira directories

Back up the Jira directories by copying them to some other location.

**✓** You can check the location of all these directories by going to **⚙️ > System > System info**, and scrolling down to the **File Paths** section.

- Back up the Jira installation directory.
- Back up the Jira home directory.
- **CLUSTER** Back up the Jira installation and home directories **on all nodes**.
- **CLUSTER** Back up the shared directory.



All good?

Head back to the landing page, and choose the upgrade method.

[Go to landing page](#)

# Upgrading Jira (installer)

You've chosen to upgrade **Jira Server or Data Center (non-clustered)** by using the **installer**.

Looking for a different upgrade method? See [Upgrading Jira applications](#).

## Skip to

- [Before you begin](#)
- [Download Jira](#)
- [Start the installation wizard](#)
- [Install the database driver](#)
- [Re-apply modifications and increase pool-max-size](#)
- [Disable automatic reindex](#)
- [Start Jira for the first time](#)
- [Post-upgrade steps](#)



Avoid using this upgrade method if you initially installed Jira manually from a .zip archive. Upgrading manual installations using the binary installer is not supported and is known to cause problems on startup.

## Before you begin

### Locate the Jira home directory to determine the initial installation method

We recommend upgrading Jira using the same method that was used to install it for the first time. If you're unsure which method that was, find the location of your Jira home directory. That's usually a good indicator of whether Jira was installed manually or from a binary installer. For more information, see [Jira application home directory](#).

### Prepare for the upgrade

Make sure you have completed the steps in [Preparing for the upgrade](#). These are mandatory pre-requisites, and are essential for a smooth upgrade.

### Choose your version

If you need help choosing the right version for you, head to the [upgrade matrix](#) to get a quick run down of features, supported platforms, and technical upgrade notes for all Jira versions.

## Download Jira

1. Download one of the Jira applications from our website. Choose the Windows or Linux installer.

- [Jira Core](#)
- [Jira Software](#)
- [Jira Service Management](#)



If you're upgrading both Jira Core/Software and Jira Service Management, upgrade Jira Core/Software only. You'll later upgrade Jira Service Management directly in Jira, without a separate installer.

## Start the installation wizard

The installation wizard will guide you through the upgrade process.

1. Run the installer you've downloaded.

a) Run the **.exe** file. We recommend using a Windows administrator account.

b) If prompted to allow the upgrade wizard to make changes to your computer, choose **Yes**. If you do not, the installation wizard will have restricted access to your operating system and any subsequent installation options will be limited.

a) Change to the directory where you downloaded Jira, then execute this command to make the installer executable:

```
$ chmod a+x atlassian-jira-X.X.X-x64.bin
```

Where **X.X.X** is the Jira version you downloaded.

b) Run the installer we recommend using `sudo` to run the installer:

```
$ sudo ./atlassian-jira-X.X.X-x64.bin
```

You can also choose to run the installer with root user privileges.

2. Follow the prompts in the wizard:

a. When prompted, choose **Upgrade an existing Jira installation**.

b. Make sure the **Existing Jira installation directory** suggested by the wizard is correct (especially important if you have multiple Jira installations on the same machine.)

c. If you have already backed up the Jira home directory, clear the **Back up the Jira home directory check box** to avoid creating an extra backup.

d. The wizard notifies you of customizations in the Jira installation directory. **Make a note of these** as you'll need to reapply them later.

 Your current customizations will be overwritten, but you can later copy them from your backups.

3. In the last screen, the upgrade wizard will ask you to start the Jira instance and complete the upgrade. **We recommend that you stop at this step**, and complete the remaining steps from this page, up until [Start Jira for the first time](#).

 Starting Jira here won't affect your upgrade in any way, but Jira needs to be shut down to complete the remaining steps. Once you complete them, you can go back to the wizard and start Jira.

4. (Optional) If you use Crowd for user management, complete these extra steps:

If you are using Crowd for user management, reapply the modifications from the following files from your existing installation directory to the new files. Do not copy the files as they may be different in the new version of Jira.

- `<Installation-Directory>/atlassian-jira/WEB-INF/classes/crowd.properties`
- `<Installation-Directory>/atlassian-jira/WEB-INF/classes/seraph-config.xml`

## Install the database driver

If you're using an **Oracle** or **MySQL** database, download a new JDBC driver. For other databases, you can omit this step.



If the driver is up to date, you can also copy it from your previous version.

1. Download one of the following drivers:
  - **Oracle:** [JDBC driver 19.3 \(ojdbc8\)](#)
  - **MySQL:** [MySQL Connector/J 5.1 driver](#)
2. Place it in `<installation-directory>/lib`.

## Step 4: Re-apply any custom changes and increase pool-max-size

While using Jira, you've probably added some custom modifications to Jira files. These may include connection details, settings related to memory allocation, or other JVM arguments.

### Migrated modifications

During the upgrade, the wizard migrated the following from your existing Jira installation:

- TCP values in the `server.xml` file.
- Location of your Jira home directory in the `jira-application.properties` file.
- The following values in the `setenv.sh/setenv.bat` file:
  - `JVM_SUPPORT_RECOMMENDED_ARGS`
  - `JVM_MINIMUM_MEMORY`
  - `JVM_MAXIMUM_MEMORY`
  - `JIRA_MAX_PERM_SIZE`

### Other modifications

Apart from the above, you need to re-apply all other modifications. Here are the most important files:

- `server.xml`
- `dbconfig.xml`
- `jira-config.properties`
- `web.xml`
- `setenv.sh/setenv.bat` (memory allocation and other JVM arguments)
- for more, see [Important files in Jira](#)

1. Re-apply your custom changes to your respective new Jira files by copying them from your backups.

Make sure you don't just copy over the old files, as the 'native' settings they contain might have changed between the Jira versions.

**i** We'll make another check on Jira startup and will show you all the files you might have skipped that still contain changes that have not been copied over. Then you'll be able to click to automatically copy the changes over.

Note that the check will only be run on the following configuration files:

- <jira-home-directory>/atlassian-jira/ directory
- <jira-home-directory>/conf/server.xml
- <jira-home-directory>/bin/setenv.sh

and the automatic transfer will only be supported for ATST plugin 1.20.0 and later.

To automatically transfer the changes, the installer copy of the modified file needs to be the same as in the version you're upgrading to.

**✓** Tomcat started to use double-quotes as of version **8.5.48** as a result of [Expansion of JAVA\\_OPTS in catalina.sh containing '\\*' stops startup on linux](#) bug. That's why when you upgrade and set parameters in setenv.sh or setenv.bat, make sure that you:

- Don't remove the double-quotes in the catalina.sh
- Set all your parameters in one line without any new line in setenv.sh or setenv.bat

Otherwise you might experience issues starting up Jira.

### Pool-max-size

If you're upgrading from Jira 7.x to Jira 8.x we recommend changing the `pool-max-size` parameter to 40 in your `dbconfig.xml` before the upgrade. Leaving the default of 20 can sometimes lead to `ResultSet Closed` errors during re-indexing on 8.x. For information on implementing the change, see [Tuning database connections](#).

## Disable automatic reindex

**i** This step is recommended for the platform upgrade, that is when upgrading from 7.x to 8.x.

Because of the changes to indexes that we've introduced in Jira 8.0, your old index is incompatible with the new version. To create a new one, Jira will trigger an automatic reindex right after you start it. To avoid reindexing twice (after startup and after upgrading your apps), you can disable the automatic reindex, and run the second one later, whenever you're ready.

1. [Edit or create](#) the following file:

```
<jira-home-directory>/jira-config.properties
```

2. Add the following line, and save the file:

```
upgrade.reindex.allowed=false
```

## Start Jira for the first time

Start your new Jira version.

1. Go back to your upgrade wizard and complete the upgrade to start Jira.

You can also start Jira by going to `<installation-directory>/bin`, and running one of the following files:

- **Windows:** `start-jira.bat`
- **Linux:** `start-jira.sh`

2. Open Jira in your browser.
3. If you've missed any file with custom changes that have not been copied over, you can automatically copy the changes over now.

Note that the check for file changes is only be run on the following configuration files:

- `atlassian-jira/` directory
- `conf/server.xml`
- `bin/setenv.sh`

and the automatic transfer is only be supported for ATST plugin 1.20.0 and later.

**i** To automatically transfer the changes, the installer copy of the modified file needs to be the same as in the version you're upgrading to.

After the changes have been copied over, you'll be prompted to restart Jira.

### Post-upgrade landing page

After a successful upgrade, you should see the post-upgrade landing page. It has some useful information about the new version, as shown below.

Welcome to Jira 8.0.0

**Latest upgrade report**

Jira was successfully upgraded to version 8.0.0 on 03/Dec/18 at 6:17 PM. Here's a summary of the checks we ran during the upgrade.

- 1 **Need to know**  
There are some important changes in this upgrade. [Show details](#)
- 2 **User apps disabled: 3**  
Some of your user apps have failed to start. They may not be compatible with this version of Jira. [Show details](#)
- 3 **Application links**  
All application links connected. [Show details](#)

For more info on the new features in Jira 8.0.0, take a look at the [Release notes](#). [Take me to Jira](#)

4

1. **Need to know:** A list of new features that might affect your work as an admin.
2. **User apps:** Status of your apps after the upgrade.

3. **Application links:**Status of your application links.
4. **Release notes:**Link to release notes where you can see more detailed information about the version you've upgraded to.

Almost there!

Your Jira instance has been upgraded. Below you can learn how to upgrade Jira Service Management, if you have it, and how to upgrade your apps.

### (Optional) Update Jira Service Management

If you're using Jira Service Management, you can update it directly in the UI, without downloading a separate installer.

1. Go to **Administration** () > **Applications > Versions and licenses**.
2. Update Jira Service Management. This will automatically update Jira Service Management to a compatible version.

### Upgrade apps (add-ons)

Now you can upgrade apps that had the **Compatible once both are updated** status. If you need more information about the statuses and apps in general, see [Preparing for the upgrade](#).

1. Go to **Administration** () > **Manage apps > Manage apps**.
2. Upgrade your apps to the supported versions.
3. Once the apps are upgraded, you can enable them.

### Rebuild index

Since your old index is incompatible, reindex Jira to rebuild it. This step might take some time, depending on how many issues and apps you have.

1. Go to **Administration** () > **Indexing**, and run **Full re-index**.

Well done!

You've upgraded Jira to a new version.

# Upgrading Jira (manual)

You've chosen to upgrade **Jira Server or Data Center (non-clustered)** manually by using the archive.

Looking for a different upgrade method? See [Upgrading Jira applications](#).

## Skip to

- [Before you begin](#)
- [Download Jira](#)
- [Extract the files](#)
- [Install the database driver](#)
- [Re-apply modification and increase max-pool-size](#)
- [Disable automatic reindex](#)
- [Start Jira for the first time](#)
- [Post-upgrade steps](#)



Avoid using this upgrade method if you initially installed Jira using the binary installer (.exe on Windows or .bin on Linux). Upgrading binary installations manually is not supported and is known to cause problems on startup.

## Before you begin

### Locate the Jira home directory to determine the initial installation method

We recommend upgrading Jira using the same method that was used to install it for the first time. If you're unsure which method that was, find the location of your Jira home directory. That's usually a good indicator of whether Jira was installed manually or from a binary installer. For more information, see [Jira application home directory](#).

### Prepare for the upgrade

Make sure you have completed the steps in [Preparing for the upgrade](#). These are mandatory pre-requisites, and are essential for a smooth upgrade.

### Choose your version

If you need help choosing the right version for you, head to the [upgrade matrix](#) to get a quick run down of features, supported platforms, and technical upgrade notes for all Jira versions.

## Download Jira

1. Download one of the Jira applications from our website.

- [Jira Software](#)
- [Jira Service Management](#) (only tar.gz archive)



If you're upgrading both Jira Software and Jira Service Management, upgrade Jira Software only. You'll later upgrade Jira Service Management directly in Jira, without a separate installer.

## Extract the files

Extract the archive you've downloaded, and start the upgrade.

1. Extract (unzip) the files to a directory (this is your new installation directory, and must be different to your existing installation directory).
2. Point Jira to your **existing** Jira home directory.

 We recommend that you do it by setting the `JIRA_HOME` environment variable. For more info on how to do this, see [Setting Jira home directory](#).

## Install the database driver

**If you're using an Oracle or MySQL database**, download a new JDBC driver. For other databases, you can omit this step.

 If the driver is up to date, you can also copy it from your previous version.

1. Download one of the following drivers:
  - **Oracle:** [JDBC driver 19.3 \(ojdbc8\)](#)
  - **MySQL:** [MySQL Connector/J 5.1 driver](#)
2. Place it in `<installation-directory>/lib`.

## Re-apply any modifications and increase max-pool-size

While using Jira, you've probably added some custom modifications to Jira files. These may include connection details, settings related to memory allocation, or other JVM arguments. In this step, you need to re-apply the same modifications to the new files by copying them from your backups.

 Make sure you don't just copy over the old files, as the 'native' settings they contain might have changed between the Jira versions.

Some of the files we usually modify:

- `server.xml`
- `dbconfig.xml`
- `jira-config.properties`
- `web.xml`
- `setenv.sh/setenv.bat` (memory allocation and other JVM arguments)
- for more, see [Important files in Jira](#)

**i** We'll make another check on Jira startup and will show you all the files you might have skipped that still contain changes that have not been copied over. Then you'll be able to click to automatically copy the changes over.

Note that the check will only be run on the following configuration files:

- <jira-home-directory>/atlassian-jira/ directory
- <jira-home-directory>/conf/server.xml
- <jira-home-directory>/bin/setenv.sh

and the automatic transfer will only be supported for ATST plugin 1.20.0 and later.

To automatically transfer the changes, the installer copy of the modified file needs to be the same as in the version you're upgrading to.

**✓** Tomcat started to use double-quotes as of version **8.5.48** as a result of [Expansion of JAVA\\_OPTS in catalina.sh containing '\\*' stops startup on linux](#) bug. That's why when you upgrade and set parameters in setenv.sh or setenv.bat, make sure that you:

- Don't remove the double-quotes in the catalina.sh
- Set all your parameters in one line without any new line in setenv.sh or setenv.bat

Otherwise you might experience issues starting up Jira.

### Pool-max-size

If you're upgrading from Jira 7.x to Jira 8.x we recommend changing the pool-max-size parameter to 40 in your dbconfig.xml before the upgrade. Leaving the default of 20 can sometimes lead to ResultSet Closed errors during re-indexing on 8.x. For information on implementing the change, see [Tuning database connections](#).

### Disable automatic reindex

**i** This step is recommended for the platform upgrade, that is when upgrading from 7.x to 8.x.

Because of the changes to indexes that we've introduced in Jira 8.0, your old index is incompatible with the new version. To create a new one, Jira will trigger an automatic reindex right after you start it. To avoid reindexing twice (after startup and after upgrading your apps), you can disable the automatic reindex, and run the second one later, whenever you're ready.

1. [Edit or create](#) the following file:

```
<jira-home-directory>/jira-config.properties
```

2. Add the following line, and save the file:

```
upgrade.reindex.allowed=false
```

### Start Jira for the first time

Start your new Jira version.

1. Go back to your upgrade wizard and complete the upgrade to start Jira.

You can also start Jira by going to `<installation-directory>/bin`, and running one of the following files:

- **Windows:** `start-jira.bat`
- **Linux:** `start-jira.sh`

2. Open Jira in your browser.
3. If you've missed any file with custom changes that have not been copied over, you can automatically copy the changes over now.

Note that the check for file changes is only be run on the following configuration files:

- `atlassian-jira/` directory
- `conf/server.xml`
- `bin/setenv.sh`

and the automatic transfer is only be supported for ATST plugin 1.20.0 and later.

**i** To automatically transfer the changes, the installer copy of the modified file needs to be the same as in the version you're upgrading to.

After the changes have been copied over, you'll be prompted to restart Jira.

### Post-upgrade landing page

After a successful upgrade, you should see the post-upgrade landing page. It has some useful information about the new version, as shown below.

Welcome to Jira 8.0.0

**Latest upgrade report**

Jira was successfully upgraded to version 8.0.0 on 03/Dec/18 at 6:17 PM. Here's a summary of the checks we ran during the upgrade.

- 1** **Need to know**  
There are some important changes in this upgrade. [Show details](#)
- 2** **User apps disabled: 3**  
Some of your user apps have failed to start. They may not be compatible with this version of Jira. [Show details](#)
- 3** **Application links**  
All application links connected. [Show details](#)

For more info on the new features in Jira 8.0.0, take a look at the [Release notes](#). [Take me to Jira](#)

**4**

1. **Need to know:** A list of new features that might affect your work as an admin.
2. **User apps:** Status of your apps after the upgrade.
3. **Application links:** Status of your application links.

4. **Release notes:** Link to release notes where you can see more detailed information about the version you've upgraded to.

Almost there!

Your Jira instance has been upgraded. Below you can learn how to upgrade Jira Service Management, if you have it, and how to upgrade your apps.

### (Optional) Update Jira Service Management

If you're using Jira Service Management, you can update it directly in the UI, without downloading a separate installer.

1. Go to **Administration** () > **Applications > Versions and licenses**.
2. Update Jira Service Management. This will automatically update Jira Service Management to a compatible version.

### Upgrade apps (add-ons)

Now you can upgrade apps that had the **Compatible once both are updated** status. If you need more information about the statuses and apps in general, see [Preparing for the upgrade](#).

1. Go to **Administration** () > **Manage apps > Manage apps**.
2. Upgrade your apps to the supported versions.
3. Once the apps are upgraded, you can enable them.

### Rebuild index

Since your old index is incompatible, reindex Jira to rebuild it. This step might take some time, depending on how many issues and apps you have.

1. Go to **Administration** () > **Indexing**, and run **Full re-index**.

Well done!

You've upgraded Jira to a new version.

# Upgrading Jira Data Center (installer)

You've chosen to upgrade **Jira Data Center (clustered)** by using the **installer**.

Looking for a different upgrade method? See [Upgrading Jira applications](#).

## Skip to

- [Before you begin](#)
- **Upgrade Jira on the first node**
  - [Download Jira](#)
  - [Start the installation wizard](#)
  - [Install the database driver](#)
  - [Re-apply modifications and increase pool-max-size](#)
  - [Disable automatic reindex](#)
- **Post-upgrade steps on the first node**
  - [Start Jira](#)
  - [Update Jira Service Management](#)
  - [Upgrade your apps](#)
  - [Rebuild index](#)
  - [Copy the upgraded Jira as a template](#)
- **Upgrade Jira on remaining nodes**



Avoid using this upgrade method if you initially installed Jira manually from a .zip archive. Upgrading manual installations using the binary installer is not supported and is known to cause problems on startup.

## Before you begin

### Step 1: Locate the Jira home directory to determine the initial installation method

We recommend upgrading Jira using the same method that was used to install it for the first time. If you're unsure which method that was, find the location of your Jira home directory. That's usually a good indicator of whether Jira was installed manually or from a binary installer. For more information, see [Jira application home directory](#).

### Step 2: Prepare for the upgrade

Make sure you have completed the steps in [Preparing for the upgrade](#). These are mandatory pre-requisites, and are essential for a smooth upgrade.

### Step 3: Choose your version

If you need help choosing the right version for you, head to the [upgrade matrix](#) to get a quick rundown of features, supported platforms, and technical upgrade notes for all Jira versions.

### Step 4: Stop the cluster



Omit this step if you're [upgrading your Data Center with zero downtime](#).

Stop Jira on all nodes in the cluster. We also recommend that you configure your load balancer to redirect the traffic away from Jira until the upgrade is complete on all nodes.

## Upgrade Jira on the first node

To avoid upgrading each of the nodes separately, you'll just upgrade one of them, and make its installation directory a **template**. Then, you'll copy this template to remaining nodes. You can choose any node here.

### Step 1. Download Jira

1. Download one of the Jira applications from our website. Choose the Windows or Linux binary.

- [Jira Software](#)
- [Jira Service Management](#)

**i** If you're upgrading both Jira Software and Jira Service Management, upgrade Jira Software only. You'll later upgrade Jira Service Management directly in Jira, without a separate installer.

### Step 2. Start the installation wizard

The installation wizard will guide you through the upgrade process.

1. Run the installer you've downloaded.

a) Run the **.exe** file. We recommend using a Windows administrator account.

b) If prompted to allow the upgrade wizard to make changes to your computer, choose **Yes**. If you do not, the installation wizard will have restricted access to your operating system and any subsequent installation options will be limited.

a) Change to the directory where you downloaded Jira, then execute this command to make the installer executable:

```
$ chmod a+x atlassian-jira-X.X.X-x64.bin
```

Where X.X.X is the Jira version you downloaded.

b) Run the installer we recommend using `sudo` to run the installer:

```
$ sudo ./atlassian-jira-X.X.X-x64.bin
```

You can also choose to run the installer with root user privileges.

2. Follow the prompts in the wizard:

a. When prompted, choose **Upgrade an existing Jira installation**.

b. Make sure the **Existing Jira installation directory** suggested by the wizard is correct (especially important if you have multiple Jira installations on the same machine.)

c. If you have already backed up the Jira home directory, clear the **Back up the Jira home directory check box** to avoid creating an extra backup.

d. The wizard notifies you of customizations in the Jira installation directory. **Make a note of these** as you'll need to reapply them later.

**i** Your current customizations will be overwritten, but you can later copy them from your backups.

- In the last screen, the upgrade wizard will ask you to start the Jira instance and complete the upgrade. **We recommend that you stop at this step**, and complete the remaining steps from this page, up until [Start Jira for the first time](#).

 Starting Jira here won't affect your upgrade in any way, but Jira needs to be shut down to complete the remaining steps. Once you complete them, you can go back to the wizard and start Jira.

### Step 3. Install the database driver

If you're using an Oracle or MySQL database, download a new JDBC driver. For other databases, you can omit this step.

 If the driver is up to date, you can also copy it from your previous version.

- Download one of the following drivers:
  - Oracle:** [JDBC driver 19.3 \(ojdbc8\)](#)
  - MySQL:** [MySQL Connector/J 5.1 driver](#).
- Place it in `<installation-directory>/lib`.

### Step 4. Re-apply modifications and increase pool-max-size

While using Jira, you've probably added some custom modifications to Jira files. These may include connection details, settings related to memory allocation, or other JVM arguments. In this step, you need to re-apply the same modifications to the new files by copying them from your backups.

 Make sure you don't just copy over the old files, as the 'native' settings they contain might have changed between the Jira versions.

#### Migrated modifications

During the upgrade, the wizard migrated the following from your existing Jira installation:

- TCP values in the `server.xml` file.
- Location of your Jira home directory in the `jira-application.properties` file.
- The following values in the `setenv.sh/setenv.bat` file:
  - `JVM_SUPPORT_RECOMMENDED_ARGS`
  - `JVM_MINIMUM_MEMORY`
  - `JVM_MAXIMUM_MEMORY`
  - `JIRA_MAX_PERM_SIZE`

#### Other modifications

Apart from the above, you need to re-apply all other modifications. Here are the most important files:

- `server.xml`
- `dbconfig.xml`
- `jira-config.properties`
- `web.xml`
- `setenv.sh/setenv.bat` (memory allocation and other JVM arguments)

- for more, see [Important files in Jira](#)

**i** We'll make another check on Jira startup and will show you all the files you might have skipped that still contain changes that have not been copied over. Then you'll be able to click to automatically copy the changes over.

Note that the check will only be run on the following configuration files:

- <jira-home-directory>/atlassian-jira/ directory
- <jira-home-directory>/conf/server.xml
- <jira-home-directory>/bin/setenv.sh

and the automatic transfer will only be supported for ATST plugin 1.20.0 and later.

To automatically transfer the changes, the installer copy of the modified file needs to be the same as in the version you're upgrading to.

### Pool-max-size

If you're upgrading from Jira 7.x to Jira 8.x we recommend changing the `pool-max-size` parameter to 40 in your `dbconfig.xml` before the upgrade. Leaving the default of 20 can sometimes lead to `ResultSet Closed` errors during re-indexing on 8.x. For information on implementing the change, see [Tuning database connections](#).

## Step 5. Disable automatic reindex

**i** This step is recommended when upgrading between platform versions. For example, when you're switching from Jira 7.x to 9.x or from Jira 7.x to 8.x.

If you're already upgrading between Jira feature releases (for example, from 8.13 to 8.20, etc.), you can omit this step.

Because of the indexing changes introduced between 8.x and 9.x Jira platform versions, indexes from any earlier Jira version will be incompatible after the upgrade.

To create a new index, Jira will trigger an automatic reindex right after you start the application. To avoid reindexing twice (after startup and after upgrading your apps), you can disable the automatic reindex and then run the second one later, whenever you're ready.

To disable automatic reindex:

1. Edit or create (if it doesn't exist) the following file:

```
<jira-home-directory>/jira-config.properties
```

2. Add the following line to the preceding file and then save your changes:

```
upgrade.reindex.allowed=false
```

---

## Post-upgrade steps on the first node

Complete these post-upgrade steps **only on the first node** (the one you've just upgraded). The remaining nodes will later download the upgraded apps and index from the shared directory.

## Step 1. Start Jira for the first time

Start your new Jira version, and connect it to the database.

1. (Installer) Go back to your upgrade wizard and start Jira.

(Installer and Manual) You can also start Jira by going to `<installation-directory>/bin`, and running one of the following files:

- **Windows:** `start-jira.bat`
- **Linux:** `start-jira.sh`

2. Open Jira in your browser.
3. Follow instructions on the screen to complete the setup.
4. If you've missed any file with custom changes that have not been copied over, you can automatically copy the changes over now.

Note that the check for file changes is only be run on the following configuration files:

- `atlassian-jira/` directory
- `conf/server.xml`
- `bin/setenv.sh`

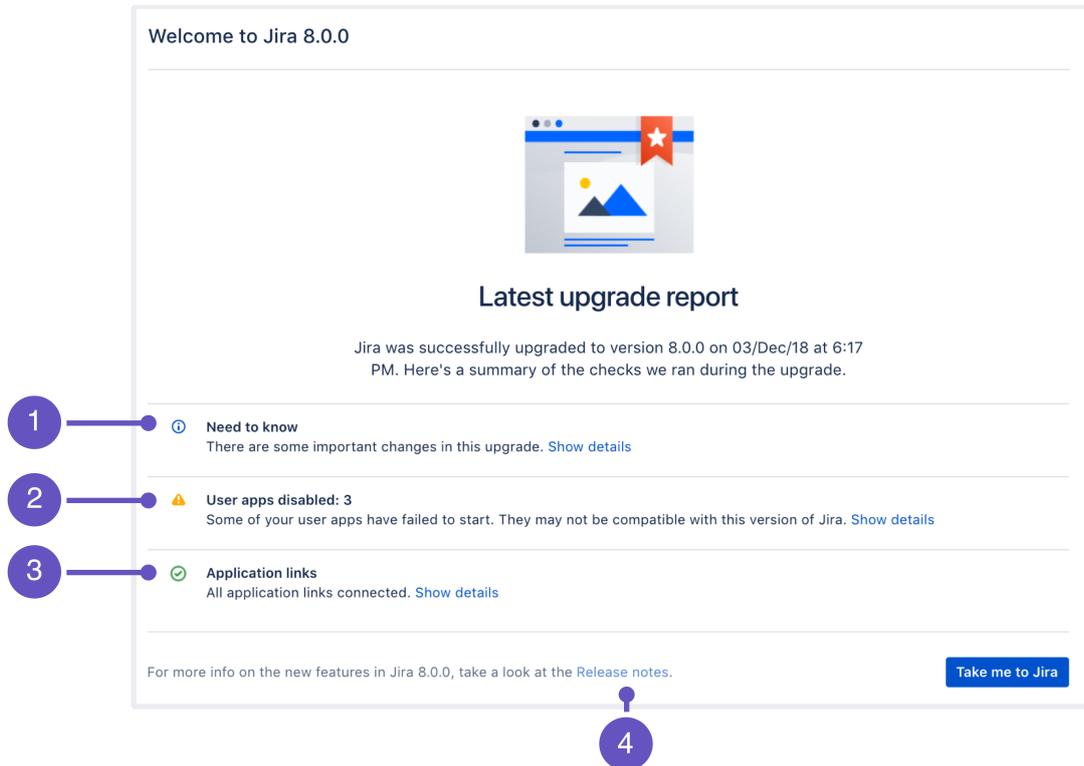
and the automatic transfer is only be supported for ATST plugin 1.20.0 and later.

 To automatically transfer the changes, the installer copy of the modified file needs to be the same as in the version you're upgrading to.

After the changes have been copied over, you'll be prompted to restart Jira.

### Post-upgrade landing page

After a successful upgrade, you should see the post-upgrade landing page. It has some useful information about the new version, as shown below.



1. **Need to know:** A list of new features that might affect your work as an admin.
2. **User apps:** Status of your apps after the upgrade.
3. **Application links:** Status of your application links.
4. **Release notes:** Link to release notes where you can see more detailed information about the version you've upgraded to.

## Step 2. (Optional) Update Jira Service Management

If you're using Jira Service Management, you can update it directly in the UI, without downloading a separate installer.

1. Go to **Administration** (⚙️) > **Applications > Versions and licenses**.
2. Update Jira Service Management. This will automatically update Jira Service Management to a compatible version.

## Step 3. Upgrade apps (add-ons)

Now you can upgrade apps that had the **Compatible once both are updated** status. If you need more information about the statuses and apps in general, see [Preparing for the upgrade](#).

1. Go to **Administration** (⚙️) > **Manage apps > Manage apps**.
2. Upgrade your apps to the supported versions.
3. Once the apps are upgraded, you can enable them.

**i** Upgrading apps in a DC cluster

When upgrading apps, each node will pull the the most recently modified app jar file from the shared home on reboot.

To determine the file version, Jira uses the <version> value inside atlassian-plugin.xml. If there are multiple <version> values, Jira uses java.lang.String#compareTo to compare the different values.

**Step 4. Rebuild index**

Reindex Jira to recreate your index. This step might take some time, depending on how many issues and apps you have.

1. Go to **Administration**  > **Indexing**, and run **Lock Jira and rebuild reindex**.

**Step 5. Copy the upgraded Jira as a template**

In this step, you'll copy the new installation directory with all the modifications you did so far. This will give you a template that you'll later copy to other nodes.

1. Copy the **new installation directory** to some other location. This will be your template.

**Upgrade remaining nodes**

By now, you should have the ready Jira template, and upgraded add-ons and index data available in the shared directory. In this step, you'll copy the template to other nodes and start them one by one.

1. Copy the **template installation directory** to the new node.
2. If the path to the local home directory is different on this node, update it in the `setenv.bat/setenv.sh` file.
3. Start Jira on this node.
4. **Rinse & repeat:** Repeat these steps on the next node.

**✔** Joining the cluster

You can check if the upgraded nodes are joining the cluster by going to **Administration**  > **System > System info**, and scrolling down to the **Cluster nodes** section.

Well done!

You've upgraded Jira to a new version.

# Upgrading Jira Data Center (manual)

You've chosen to upgrade **Jira Data Center (clustered) manually** by using the archive.

Looking for a different upgrade method? See [Upgrading Jira applications](#).

## Skip to

- [Before you begin](#)
- **Upgrade Jira on the first node**
  - [Download Jira](#)
  - [Extract the files](#)
  - [Install the database driver](#)
  - [Re-apply modifications and increase pool-max-size](#)
  - [Disable automatic reindex](#)
- **Post-upgrade steps on the first node**
  - [Start Jira for the first time](#)
  - [Update Jira Service Management](#)
  - [Upgrade your apps](#)
  - [Rebuild index](#)
  - [Copy the upgraded Jira as a template](#)
- **Upgrade Jira on remaining nodes**



Avoid using this upgrade method if you initially installed Jira using the binary installer (.exe on Windows or .bin on Linux). Upgrading binary installations manually is not supported and is known to cause problems on startup.

## Before you begin

### Step 1: Locate the Jira home directory to determine the initial installation method

We recommend upgrading Jira using the same method that was used to install it for the first time. If you're unsure which method that was, find the location of your Jira home directory. That's usually a good indicator of whether Jira was installed manually or from a binary installer. For more information, see [Jira application home directory](#).

### Step 2: Prepare for the upgrade

Make sure you have completed the steps in [Preparing for the upgrade](#). These are mandatory pre-requisites, and are essential for a smooth upgrade.

### Step 3: Choose your version

If you need help choosing the right version for you, head to the [upgrade matrix](#) to get a quick rundown of features, supported platforms, and technical upgrade notes for all Jira versions.

### Step 4: Stop the cluster



Omit this step if you're [upgrading your Data Center with zero downtime](#).

Stop Jira on all nodes in the cluster. We also recommend that you configure your load balancer to redirect the traffic away from Jira until the upgrade is complete on all nodes.

## Upgrade Jira on the first node

To avoid upgrading each of the nodes separately, you'll just upgrade one of them, and make it a **template**. Then, you'll copy this template to remaining nodes. You can choose any node here.

### Step 1. Download Jira

1. Download one of the Jira applications from our website. Choose the tar.gz or zip archive.

- [Jira Software](#)
- [Jira Service Management](#) (only tar.gz archive)

 If you're upgrading both Jira Software and Jira Service Management, upgrade Jira Software only. You'll later upgrade Jira Service Management directly in Jira, without a separate installer.

### Step 2. Extract the files

Extract the archive you've downloaded, and start the upgrade.

1. Extract (unzip) the files to a directory (this is your new installation directory, and must be different to your existing installation directory).
2. Point Jira to your **existing** Jira home directory.

 We recommend that you do it by setting the `JIRA_HOME` environment variable. For more info on how to do this, see [Setting Jira home directory](#).

### Step 3. Install the database driver

**If you're using an Oracle or MySQL database**, download a new JDBC driver. For other databases, you can omit this step.

 If the driver is up to date, you can also copy it from your previous version.

1. Download one of the following drivers:
  - **Oracle:** [JDBC driver 19.3 \(ojdbc8\)](#)
  - **MySQL:** [MySQL Connector/J 5.1 driver](#).
2. Place it in `<installation-directory>/lib`.

### Step 4: Re-apply any custom changes and increase pool-max-size

While using Jira, you've probably added some custom modifications to Jira files. These may include connection details, settings related to memory allocation, or other JVM arguments. Usually, these are the files that contain custom changes:

- `server.xml`
- `dbconfig.xml`
- `jira-config.properties`
- `web.xml`

- for more, see [Important files in Jira](#).

1. Re-apply your custom changes to your respective newJira files by copying them from your backups.

Make sure you don't just copy over the old files, as the 'native' settings they contain might have changed between the Jira versions.

 We'll make another check on Jira startup and will show you all the files you might have skipped that still contain changes that have not been copied over. Then you'll be able to click to automatically copy the changes over.

Note that the check will only be run on the following configuration files:

- <jira-home-directory>/atlassian-jira/ directory
- <jira-home-directory>/conf/server.xml
- <jira-home-directory>/bin/setenv.sh

and the automatic transfer will only be supported for ATST plugin 1.20.0 and later.

To automatically transfer the changes, the installer copy of the modified file needs to be the same as in the version you're upgrading to.

 Tomcat started to use double-quotes as of version **8.5.48** as a result of [Expansion of JAVA\\_OPTS in catalina.sh containing '\\*' stops startup on linux](#) bug. That's why when you upgrade and set parameters in setenv.sh or setenv.bat, make sure that you:

- Don't remove the double-quotes in the catalina.sh
- Set all your parameters in one line without any new line in setenv.sh or setenv.bat

Otherwise you might experience issues starting up Jira.

### Pool-max-size

If you're upgrading from Jira 7.x to the next platform version, such as 8.x or 9.x, we recommend that you change the pool-max-size parameter to 40 in your dbconfig.xml before the upgrade. Leaving the default value of 20 can cause various issues related to database connection. For information on implementing the change, see [Tuning database connections](#).

## Step 5. Disable automatic reindex

 This step is recommended when upgrading between platform versions. For example, when you're switching from Jira 7.x to 9.x or from Jira 7.x to 8.x.

If you're already upgrading between Jira feature releases (for example, from 8.13 to 8.20, etc.), you can omit this step.

Because of the indexing changes introduced between 8.x and 9.x Jira platform versions, indexes from any earlier Jira version will be incompatible after the upgrade.

To create a new index, Jira will trigger an automatic reindex right after you start the application. To avoid reindexing twice (after startup and after upgrading your apps), you can disable the automatic reindex and then run the second one later, whenever you're ready.

To disable automatic reindex:

1. Edit or create (if it doesn't exist) the following file:

```
<jira-home-directory>/jira-config.properties
```

2. Add the following line to the preceding file and then save your changes:

```
upgrade.reindex.allowed=false
```

---

## Post-upgrade steps on the first node

Complete these post-upgrade steps **only on the first node** (the one you've just upgraded). The remaining nodes will later download the upgraded apps and index from the shared directory.

### Step 1. Start Jira for the first time

Start your new Jira version, and connect it to the database.

1. (Installer) Go back to your upgrade wizard and start Jira.

(Installer and Manual) You can also start Jira by going to `<installation-directory>/bin`, and running one of the following files:

- **Windows:** `start-jira.bat`
- **Linux:** `start-jira.sh`

2. Open Jira in your browser.
3. Follow instructions on the screen to complete the setup.
4. If you've missed any file with custom changes that have not been copied over, you can automatically copy the changes over now.

Note that the check for file changes is only be run on the following configuration files:

- `atlassian-jira/` directory
- `conf/server.xml`
- `bin/setenv.sh`

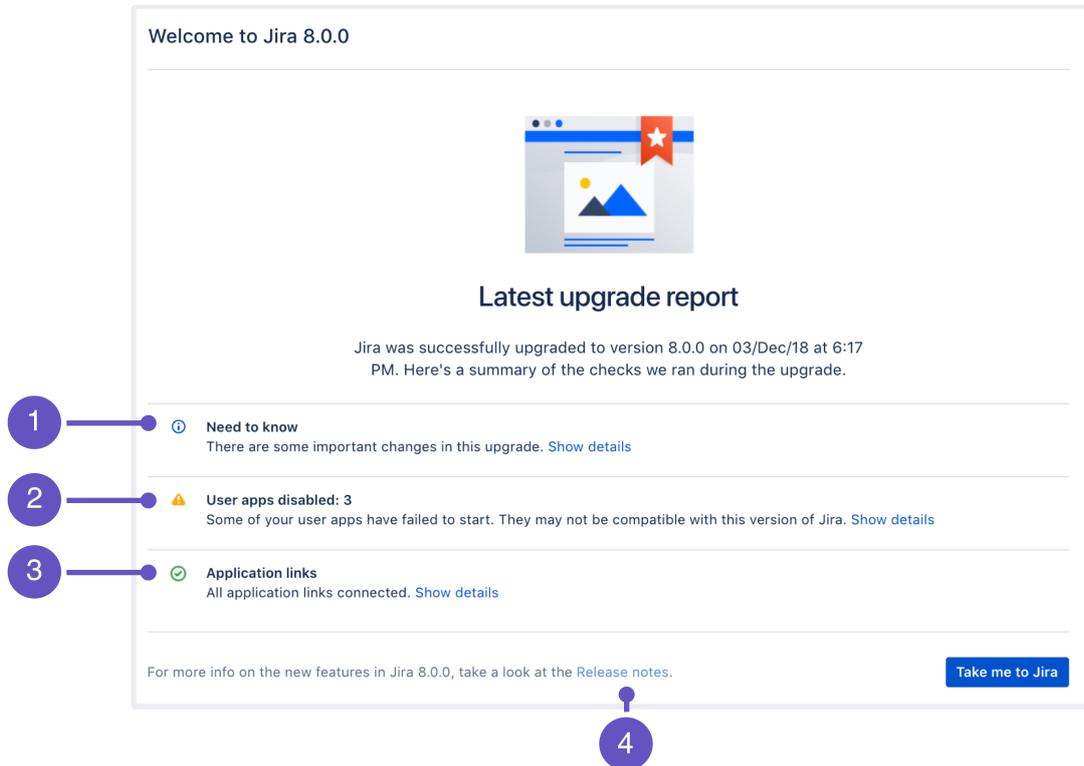
and the automatic transfer is only be supported for ATST plugin 1.20.0 and later.

 To automatically transfer the changes, the installer copy of the modified file needs to be the same as in the version you're upgrading to.

After the changes have been copied over, you'll be prompted to restart Jira.

### Post-upgrade landing page

After a successful upgrade, you should see the post-upgrade landing page. It has some useful information about the new version, as shown below.



1. **Need to know:** A list of new features that might affect your work as an admin.
2. **User apps:** Status of your apps after the upgrade.
3. **Application links:** Status of your application links.
4. **Release notes:** Link to release notes where you can see more detailed information about the version you've upgraded to.

## Step 2. (Optional) Update Jira Service Management

If you're using Jira Service Management, you can update it directly in the UI, without downloading a separate installer.

1. Go to **Administration** (⚙️) > **Applications > Versions and licenses**.
2. Update Jira Service Management. This will automatically update Jira Service Management to a compatible version.

## Step 3. Upgrade apps (add-ons)

Now you can upgrade apps that had the **Compatible once both are updated** status. If you need more information about the statuses and apps in general, see [Preparing for the upgrade](#).

1. Go to **Administration** (⚙️) > **Manage apps > Manage apps**.
2. Upgrade your apps to the supported versions.
3. Once the apps are upgraded, you can enable them.

**i** Upgrading apps in a DC cluster

When upgrading apps, each node will pull the the most recently modified app jar file from the shared home on reboot.

To determine the file version, Jira uses the <version> value inside atlassian-plugin.xml. If there are multiple <version> values, Jira uses `java.lang.String#compareTo` to compare the different values.

**Step 4. Rebuild index**

Reindex Jira to recreate your index. This step might take some time, depending on how many issues and apps you have.

1. Go to **Administration**  > **Indexing**, and run **Lock Jira and rebuild reindex**.

**Step 5. Copy the upgraded Jira as a template**

In this step, you'll copy the new installation directory with all the modifications you did so far. This will give you a template that you'll later copy to other nodes.

1. Copy the **new installation directory** to some other location. This will be your template.

**Upgrade remaining nodes**

By now, you should have the ready Jira template, and upgraded add-ons and index data available in the shared directory. In this step, you'll copy the template to other nodes and start them one by one.

1. Copy the **template installation directory** to the new node.
2. If the path to the local home directory is different on this node, update it in the `setenv.bat/setenv.sh` file.
3. Start Jira on this node.
4. **Rinse & repeat:** Repeat these steps on the next node.

**✔** Joining the cluster

You can check if the upgraded nodes are joining the cluster by going to **Administration**  > **System > System info**, and scrolling down to the **Cluster nodes** section.

Well done!

You've upgraded Jira to a new version.

# Upgrading Jira Data Center with zero downtime

You've chosen to upgrade **Jira Data Center (clustered)** with **zero downtime**.

 Zero downtime upgrade is not available when upgrading from Jira 7.x to Jira 8.x. You'll need to use one of the [regular methods](#). If you're already on Jira 8.x, you can use zero downtime to upgrade to any later version.

## Skip to

- [Before you begin](#)
- [Put Jira into upgrade mode](#)
- [Update Jira Service Management](#)
- [Upgrade Jira](#)
- [Finalize your upgrade](#)

## About Zero downtime upgrades

Upgrade with zero downtime is a special method available for Jira Data Center. It introduces the *upgrade mode* that allows your nodes to work on different Jira versions while you upgrade them one by one. During the upgrade, Jira remains fully functional and open to your users.

You can use this method to upgrade Jira Software Data Center and Jira Service Management Data Center. Jira Software 7.3 or Jira Service Management 3.6 are the *minimum* versions you need to be able to use this upgrade process. If you're running a Jira installation with both Jira Software and Jira Service Management, don't worry, we got you covered!

As the zero downtime upgrade process can be quite lengthy (depending on how many nodes you have), we've also got a [handy checklist](#) you can use to make sure you've done everything you need to. We still recommend you go through all the steps on this page, the checklist is just a handy tool to help keep track of what you're doing.

## Technical overview

For more information on what happens with your cluster in terms of upgrade, see [ZDU technical overview](#).

## FAQs

If you still have doubts, have a look at our [Zero downtime upgrade FAQs](#).

## Before you begin

### Step 1: Prepare for the upgrade

Make sure you have completed the steps in [Preparing for the upgrade](#). These are mandatory pre-requisites, and are essential for a smooth upgrade.

You must also check that a [data pipeline export](#) is not in progress.

### Step 2: Choose your version

If you need help choosing the right version for you, head to the [upgrade matrix](#) to get a quick run down of features, supported platforms, and technical upgrade notes for all Jira versions.

## PutJira into upgrade mode

PutJira into upgrade mode to allow your nodes to work on different versions while you upgrade them one by one.

1. Go to **Administration** (⚙️) > **Applications > Jira upgrades**.
2. Click **PutJira into upgrade mode**. This will only be available if your nodes are all on the same version.

### Canceling the upgrade

- You have the option to cancel the upgrade, which will take Jira out of the upgrade mode, until you start upgrading your nodes. The option will be disabled then.
- To cancel the upgrade later, you'd need to roll each node back to its original version.

## UpdateJira Service Management

This step is required only if you use both Jira Software and Jira Service Management. If you use only one of them, you can omit this step.

1. Download the required [Jira Service Management OBR](#) file. Make sure you download the version that is compatible with the Jira Software version you're going to install. The compatible version is listed next to Jira Service Management's version, for example *Jira Server 7.12.3*.
2. Change the extension of the OBR file you downloaded from `.obr` to `.zip`.
3. Unzip this file to extract the contents.
4. Copy all the jar files from the directory where you extracted the contents of the zip file and from the child directory "dependencies" and place them in `<Jira shared home>/plugins/installed-plugins` (Read more about the shared home directory in step 2 [here](#)).

The terminal commands are:

#### Linux

```
cp *.jar dependencies/*.jar <Jira shared home>/plugins/installed-plugins
```

#### Windows

```
copy *.jar + dependencies/*.jar <Jira shared home>/plugins/installed-plugins
```

During the upgrade process, upgraded nodes will pick up the new Jira Service Management jar files from the shared home, while nodes that haven't been upgraded will continue using the old versions of jars. When your upgrade is complete, all your nodes will be running the new version of Jira Service Management.

## UpgradeJira

Once your Jira instance is in *upgrade mode*, you can upgrade each node individually. Upgrading a node will involve stopping Jira, upgrading the installation, and then starting Jira. Stopping Jira will remove the node from your cluster, making it unavailable, and any users logged in to that node will lose their current session, before being routed to another node. As the administrator, it's up to you to decide which nodes to upgrade and in which order. You always need to have at least one node online and connected to your cluster to achieve zero downtime.

- [DATA CENTER](#) [Upgrading Jira Data Center \(installer\)](#)
- [DATA CENTER](#) [Upgrading Jira Data Center \(manual\)](#)

## Finalize your upgrade

Finalizing an upgrade will allow any required upgrade tasks to run on your instance, and take Jira out of upgrade mode. Once the required tasks have completed, your installation is upgraded.

1. Navigate to **Administration**  **> Applications > Jira upgrades**.
2. Click **Finalize upgrade**. This will only be available if all your nodes are all on the same, new version.

## In case you need to roll back

As a sys admin you can cancel the upgrade by clicking the **Cancel upgrade** button at any time during the upgrade as long as you have not clicked **Finalize upgrade**. If you cancel the upgrade, you can restart it at any time.

After cancelling, you need to stop each node that has been upgraded, re-install the original version on these nodes and add them again to the cluster.

# Zero downtime upgrade checklist

When you're performing a zero downtime upgrade on a Jira Software or Jira Service Management Data Center instance, it's important that you complete each step and verify it's been successful. You can use this checklist to make sure that you've literally ticked all the boxes.

## Before you begin

- We strongly recommend performing the upgrade in a test environment first.
- Ensure you have the installer for your intended version.
- Ensure you know about your intended version by reviewing the release notes.
- Ensure you're on a [supported platform](#).
- Ensure you check your add-ons.
- Back up your database using your database's native tools.
- Ensure your Support Healthcheck, Instance Health, and [Troubleshooting and Support Tools](#) plugins are enabled, and up to date with the latest versions.
- Ensure autoscaling is disabled if running in AWS or any other private cloud environment that supports autoscaling

## Before upgrading a node

- Ensure you have initiated ZDU (cluster is in 'Ready to upgrade' mode).
- Ensure no long-running tasks are active.

## Before running upgrade tasks

- Ensure all nodes have rejoined the cluster on the new version.
- Ensure all nodes load Jira as expected.
- Run your own smoke tests or test suite to ensure all works as expected.
- Back up your database.

Download the checklist so that you have it available for every node.

[Download checklist](#)

## Upgrade task troubleshooting

When you upgrade Jira Software or Jira Service Management Data Center using zero downtime upgrades, and depending on what version you're upgrading to, your instance may need to run upgrade tasks on your data. Occasionally, these upgrade tasks will fail to complete, or fail to run correctly. This can be for a variety of reasons. For general troubleshooting, we suggest you review your logs to locate the potential problem.

# Zero downtime upgrade FAQs

Have a burning question about zero downtime upgrade (ZDU)? We've gathered here the most frequently asked questions, so your upgrade goes smoothly.

[Supported versions](#)

[Database](#)

[Nodes](#)

[Performance](#)

[Reindexing](#)

[Apps](#)

[Troubleshooting](#)

## Supported versions

No. Zero downtime upgrade (ZDU) is available only for the enterprise editions of Jira, **Jira Software Data Center** and **Jira Service Management Data Center**, versions 7.3 or later.

Yes, you can upgrade to any Jira version after 7.3, as long as you stay within the same platform release, like the 7.x line.

When you choose to skip multiple feature release versions (e.g. 7.3 to 7.6), make sure to test the upgrade in a staging environment. With every release, Atlassian tests the upgrade path from previous version to newest version (upgrading a single feature release version only). Zero downtime upgrade (ZDU), however, doesnt have any constraints that wouldnt let it you skip multiple versions during an upgrade.

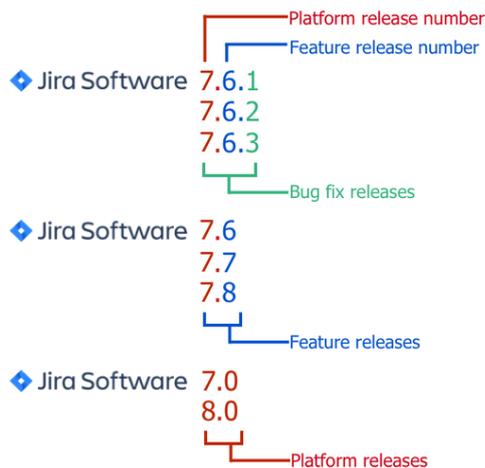
Yes, you can upgrade to any Jira version after 7.3, as long as you stay within the same platform release, for example the 7.x line.

When you choose to skip multiple versions (e.g. 7.3 to 7.6), make sure to test the upgrade in a staging environment. With every release, were testing the upgrade from previous version to current version (upgrading through a single version only), as its impossible to test all possible combinations. Zero downtime upgrade (ZDU), however, doesnt have any constraints that wouldnt let it go through multiple versions.

No. Zero downtime upgrade (ZDU) isnt supported for upgrading between **platform releases** (e.g. between the 7.x line and the 8.x line).

Thats because platform releases introduce fundamental changes to how Jira works, and require a normal upgrade. Once you move to the new platform release line, you can use zero downtime upgrade to upgrade to this platforms feature and bug fix releases (e.g. from 8.0 to 8.1).

Heres an overview of the differences between platform, feature, and bug fix releases:



## Database

Yes; just like with other upgrade methods, you should always back up your database prior to the upgrade, preferably using the database native tools.

Yes, you can roll back to your previous version at any point before **finalizing the upgrade** (its the last step of the upgrade process).

For the database, the database schema will be updated after you upgrade the first node, and these changes wont be reverted. This shouldnt be a problem, though, as schema changes are backward compatible and can work with your previous version.

## Nodes

After **finalizing the upgrade**, a non-upgraded node wont be allowed to join the cluster, and will be locked on startup. In a case like this, you can shut it down, upgrade the node, and then attempt to rejoin it with the rest of the Data Center cluster.

## Performance

During the upgrade, each node will be shut down (one at a time), upgraded, and then started again. Once a particular node is unavailable, the traffic and the load will shift to remaining nodes, which could affect performance. We recommend that you dont schedule your zero downtime upgrade at the same time as your peak traffic hours. However, you can still expect that you will have continuity of service when you choose to run your upgrade using ZDU.

## Reindexing

No. In feature and bug fix releases, where zero downtime upgrade (ZDU) is supported, new versions dont introduce changes that would require an automatic re-index.

## Apps

We recommend upgrading apps separately, after the Jira Data Center cluster has been upgraded.

During the zero downtime upgrade, Jira freezes all apps for nodes that are still running the original version of Jira. This means that even if you upgrade apps during the upgrade, all Jira nodes that werent yet upgraded will continue to run the old version of the apps. Once you upgrade a particular node, it will immediately pick up the upgraded, compatible apps.

If youre an app developer and need more details, go [here](#).

## Troubleshooting

Yes, you can roll back to your previous version at any point before **finalizing the upgrade** (its the last step of the upgrade process).

For the database, the database schema will be updated after you upgrade the first node, and these changes won't be reverted. This shouldn't be a problem, though, as schema changes are backward compatible and can work with your previous version.

Yes. We're tracking bugs that might affect ZDU on our public Jira instance, all labeled with *affects-zdu*. [You can view them here](#)

# Upgrading Jira with a fallback method

You've chosen to upgrade **Jira Server or Data Center** with a **fallback method**.

## Skip to

- [About the fallback method](#)
- [Set up a proxy server](#)
- [Shut down Jira and create backups](#)
- [Set up a new Jira instance](#)
- [Upgrade Jira](#)
- [Verify the upgrade and redirect the proxy](#)

## About the fallback method

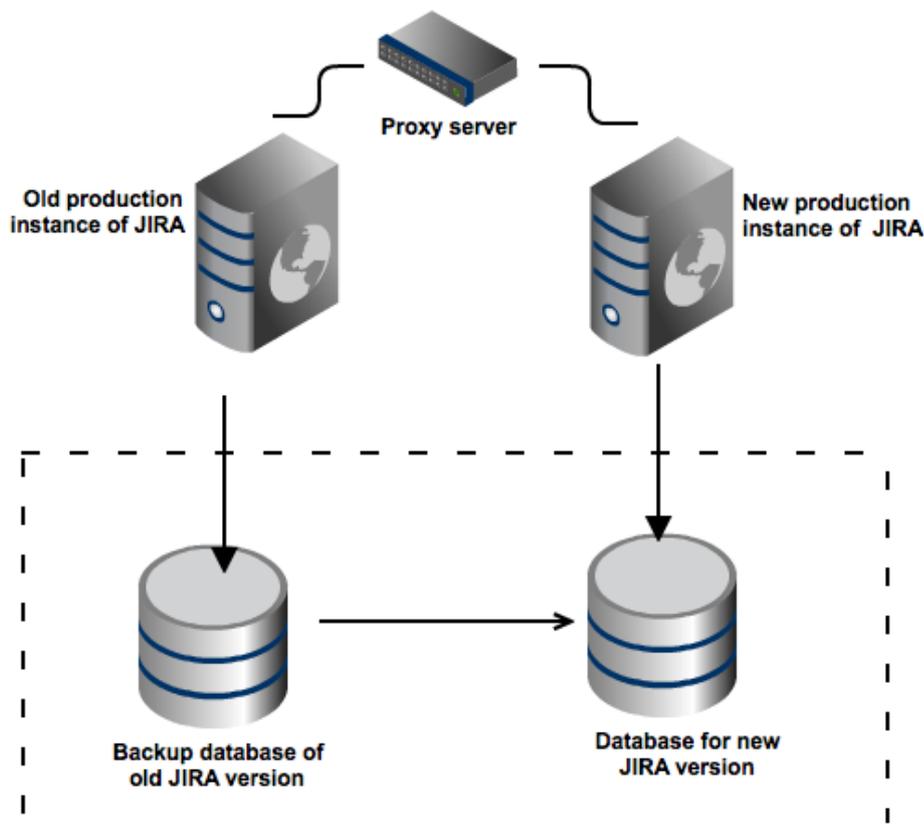
The fallback method allows you safely and quickly roll back to your previous version if the upgrade process takes longer than expected, or if you encounter any issues.

This method is especially useful for:

- Enterprise environments where Jira is mission-critical for the business, and you can't allow prolonged downtime.

To be able to quickly change the Jira instance your users are being redirected to, you'll need to set up a proxy server. By doing so, you'll have a quick way of redirecting them either to the new instance of Jira (if you're happy with the upgrade), or to the old one (if something went wrong). Your current Jira environment will be left untouched, and you'll be replicating it to a new location.

This graphic illustrates the process described in this document. For simplicity, the illustration shows how you can perform an upgrade using two different pieces of hardware. However, you can just as easily install Jira in different directories on the same server to test and perform an upgrade. In this case, simply ensure that you use separate installation and database directories during the testing.



## Set up a proxy server

Set up a reverse proxy, such as a load balancer. The proxy server lets you redirect users to a different Jira server without having to wait for a DNS change. If, at any point during the upgrade process, you encounter issues you can't resolve, you can restart your existing Jira instance and reconfigure the proxy server to point to the old Jira instance.

If you use monitoring, API calls (such as SOAP, REST, or CLI), or scripts associated with your production server, update them with the new proxy information.

Please see the following documentation for further information on configuring Apache:

- [Integrating Jira with Apache](#)
- [Integrating Jira with Apache using SSL](#)

## Shut down Jira and create backups

Shut down your existing Jira, so that users don't create new data. Next, create backups of your database, and the home and installation directories.

1. Back up your database and confirm the backup was created properly.
2. Back up your **installation directory** and **home directory**.

## Set up a new Jira instance

The easiest way to set up a new environment is to use the procedure described in [Creating a test environment for Jira](#), where you copy the whole Jira into a new directory.



You can also just install the new Jira instance from scratch, and then restore the database, home directory, and all your customizations. However, replicating your old instance might be a better idea, because you'll be able to test the upgrade on it.

## Upgrade Jira

Perform a regular upgrade on your new Jira instance. Whether you choose the installer or manual is up to you.

- [Upgrading Jira \(installer\)](#)
- [Upgrading Jira \(manual\)](#)
- CLUSTER [Upgrading Jira Data Center \(installer\)](#)
- CLUSTER [Upgrading Jira Data Center \(manual\)](#)

## Verify the upgrade and redirect the proxy

Take a look around your new instance and verify that everything is working properly:

- **All good**  
If you're happy with the outcome, redirect the proxy server to the new Jira instance.
- **Not really**  
If something is not right, redirect the proxy server to the old Jira instance. Your users can resume work, while you prepare for the new upgrade.

# Rolling back a Jira application upgrade

You can roll back Jira to its previous version if you encounter any issues with the upgrade. Any data changed since the last backup will not be present after rolling back.

## Before you begin

Make sure you have the following backups from your previous version:

- The Jira database (created with the database's native tools)
- The Jira home directory
- The Jira installation directory

## Rolling back the upgrade

To roll back the upgrade, you simply need to restore the database from the backup and copy the installation and home directories to their original locations, like it was in your previous Jira setup. Once you do it and restart Jira, your old environment will be restored.

1. Stop the upgrade or the upgraded Jira instance.
2. Use your database tools to restore the Jira database from the backup.
3. Restore the Jira **installation directory** to its original location. In the case of a manual upgrade, the original installation directory should be intact.
4. Restore the Jira **home directory** to its original location.
5. Start Jira by running the `start-jira.sh` or `start-jira.bat` file in the `bin` subdirectory of your restored Jira application installation directory.

If that's the case, you also need to restart the `AtlassianJira` service from the Control Panel. No need to create a new one, since the Jira service entry is retained even if there's an error during the upgrade to facilitate the rollback.

**Well done! You've rolled back Jira to its previous version.**

# Establishing staging server environments for Jira applications

## Redirection Notice

This page will redirect to [ADMINJIRASERVER:Creating a test environment for Jira.](#)

# Migrating Jira applications to another server

This document describes how to migrate/upgrade to Jira applications on different server hardware, or in a different server environment that entails one or more of the following:

- new operating system
- new locations for storing your index and/or attachments
- new database or database system

To migrate Jira to a new server or location, you'll need to install a new Jira instance. Once you've completed the installation, you'll migrate your existing data between the databases, and then move your home directory and all existing customizations.

## Jira Data Center Migration App

If you're planning to migrate your Jira instance to clustered infrastructure on AWS, consider using the [Jira Data Center Migration App](#). This app will automate the customization and deployment of AWS infrastructure; afterwards, it'll also walk you through the migration process. For detailed information, see [How to use the Data Center Migration app to migrate Jira to an AWS cluster](#).

## Back up

1. Create a backup of your database. For details, see.
  - a. [Stop users from updating data in Jira](#).
  - b. [Backing up data](#)
2. Back up your installation and home directory.
3. As soon as the backup is complete, delete the `<home-directory>/dbconfig.xml` file from your existing home directory. This will remove the existing connection between Jira and your database, and allow you to connect to a brand new database.

## Download Jira

1. Download one of the Jira applications from our website. Choose the zip or tar.gz archive.
  - [Jira Core](#)
  - [Jira Software](#)

 If you're upgrading both Jira Core/Software and Jira Service Desk, upgrade Jira Core/Software only. You'll later upgrade Service Desk directly in Jira, without a separate installer.

## Extract the files

Extract the archive you've downloaded, and start the upgrade.

1. Stop your **existing Jira instance**.
2. Extract (unzip) the files to a directory (this is your new installation directory, and must be different to your existing installation directory).
3. Edit the following file:

```
<installation-directory>\atlassian-jira\WEB-INF\classes\jira-application.properties
```

It must point to your **existing Jira** home directory. Make sure that you deleted the `dbconfig.xml` file, otherwise Jira will try to connect to your existing database.

4. (Optional) If you use Crowd for user management, complete these extra steps.

If you are using Crowd for user management, reapply the modifications from the following files from your existing installation directory to the new files. Do not copy the files as they may be different in the new version of Jira.

- `<Installation-Directory>/atlassian-jira/WEB-INF/classes/crowd.properties`
- `<Installation-Directory>/atlassian-jira/WEB-INF/classes/seraph-config.xml`

## Install the database driver

If you're using an **Oracle** or **MySQL** database, download a new JDBC driver. For other databases, you can omit this step.

 If the driver is up to date, you can also copy it from your previous version.

1. Download one of the following drivers:

- **Oracle:**JDBC driver 12.2.0.1.
- **MySQL:**the latest JDBC driver.

2. Place it in `<installation-directory>/lib`.

## Re-apply any modifications

While using Jira, you've probably added some custom modifications to Jira files. These may include connection details, settings related to memory allocation, or other JVM arguments. In this step, you need to re-apply the same modifications to the new files by copying them from your backups.

 Make sure you don't just copy over the old files, as the 'native' settings they contain might have changed between the Jira versions.

Some of the files we usually modify:

- `server.xml`
- `dbconfig.xml`
- `Jira-config.properties`
- `setenv.sh/setenv.bat`(memory allocation and other JVM arguments)
- for more, see [Important files in Jira](#)

## Start Jira for the first time

Start your new Jira version, and connect it to the database.

1. Go to `<installation-directory>/bin`, and run one of the following files:

- **Windows:** `start-jira.bat`
- **Linux:** `start-jira.sh`

2. Open Jira in your browser.
3. When prompted, choose **I'll set it up myself** to get access to more setup options.
4. Select **My own database**, and provide details of a **new, empty database**.
5. On the next screen, click **Import your data**, and select the file with your XML backup.

### Set up application properties

---

**Existing data?** You can [import your data](#) from another installed or hosted Jira server instead of completing this setup process.

Application Title

The name of this installation.

Mode  **Private**  
Only administrators can create new users.

**Public**  
Anyone can sign up to create issues.

Base URL

The base URL for this installation of Jira.  
All links created will be prefixed by this URL.

---

[Next](#)

 Avoid passing through a proxy when importing your XML data, especially if your Jira instance is large. Using a proxy may cause timeout errors.

6. Follow instructions on the screen to complete the setup.

## Restore your backup and attachments

Time to unpack your backup. For instructions, see [Restoring data](#).

### Almost there!

Your Jira instance has been migrated. Below you can learn how to upgrade Service Desk, if you have it, and how to upgrade your apps.

### (Optional) Update Jira Service Desk

If you're using Jira Service Desk, you can update it directly in the UI, without downloading a separate installer.

1. Go to **Administration** () > **Applications > Versions and licenses**.
2. Update Jira Service Desk. This will automatically update Service Desk to a compatible version.

### Upgrade apps (add-ons)

Upgrade your apps, so they're compatible with the new version.

1. Go to **Administration** ()  > **Manage apps > Manage apps**.

2. Upgrade your apps to the supported versions.

### Rebuild index

Your old index is incompatible with the new version, and will be deleted by Jira. Run the reindex to rebuild it from scratch. This step might take some time, depending on how many issues and apps you have.

1. Go to **Administration** () > **Indexing**, and run **LockJira and rebuild index**.

# Migrating from Jira Cloud to Server applications

## **i** Important changes to our server and Data Center products

We've ended sales for new Server licenses, and will end support for server on February 15, 2024. We're continuing our investment in Data Center with several key improvements. [Learn what this means for you](#)

**i** This page is about moving data from Jira Cloud to Jira Server. If you want to migrate from Jira Server to Jira Cloud, [check out our migration resources](#).

Jira Cloud is typically ahead of Jira Server, which means that some features may not be available after you've moved to Jira Server.

If you want to move a project, not your entire site, see [Restoring a project from backup](#).

## Limitations

Jira Cloud applications are regularly updated with the latest features and improvements, which means they are a later version than the latest downloadable version of Jira applications. If you want to migrate from Jira Cloud applications to Jira Server applications, be aware of the following limitations.

### Feature loss

Jira Cloud typically contains features that are not yet released in the latest version of Jira Server.

### Password reset

Your Jira Cloud users will need to reset their passwords before they can log in to the new Jira Server instance.

### Jira application licenses

Your Atlassian Cloud license can't be used in an instance installed from Jira Server applications. You'll need to generate a new Jira Server application license from <https://my.atlassian.com>.

You can reuse your licenses for plugins in your instance installed from Jira Server applications. The licenses for Atlassian plugins and Gliffy for Jira applications can be viewed on <https://my.atlassian.com>. For all other third-party plugins, contact the third-party vendor for a license.

### User avatars

Due to recent changes that took place in Cloud, User avatars now reside in [id.atlassian.com](https://id.atlassian.com). They are not included in the Cloud backup and, therefore, cannot be migrated to JIRA Server.

### Migrating other Cloud applications

The instructions on this page only apply to Jira applications. If you are migrating other Cloud applications (e.g. Confluence Cloud to an instance installed from Confluence Server), see this page: [Backing up and exporting data](#).

Note, if you are migrating Jira Cloud applications and other applications (e.g. Confluence Cloud) to an instance hosted on your own servers, you'll lose integration features that are native to Cloud. These can be re-enabled by configuring application links between your applications. See [Using AppLinks to link to other applications](#) for instructions. Contact support if you need assistance.

## Migrating to Jira Server

### 1. Back up

Back up your Jira Cloud application data.

1. Log in to Jira Cloud as an administrator.
2. Create an XML backup. For more info, see [Exporting issues from Cloud to Server](#). Note that the export process will strip your cloud application and plugin licenses out of the XML, so the licenses will remain available in your Jira Cloud site but not in your Server instance.

## 2. Download the installer

Download the installer for your operating system [Jira Core](#), [Jira Software](#), or [Jira Service Desk](#).

## 3. Install Jira Server

Install your Jira applications. For detailed instructions, refer to [Installing Jira applications](#).

Note that during the installation, you will be asked if you have existing data. This is where you can import your XML backup, as described in the next step.

## 4. Import your data from Jira Cloud into Jira Server

In step 2 of the setup wizard (Application Properties), you'll be asked whether you have existing data. Click **Import your existing data**, and follow instructions to import your XML backup.

When importing, you might see a warning that you're importing data from an earlier Jira version. You can ignore it and continue with the import.

### If your backup is 2 GB or more, import the attachments separately

For large backups, we recommend that the attachments are imported separately. To do this:

1. Unzip the backup file.
2. Compress the `activeobjects.xml` and the `entities.xml` files only.
3. Import the compressed file in the setup wizard, as described above.
4. Copy the contents of the `attachments` directory into the `<home-directory>/data/attachments` directory for Jira Server.

## After the migration

### 5. Change the admin password

1. Log in to your new Jira application, using the following credentials:
  - Username: `sysadmin`
  - Password: `sysadmin`
2. Change the password immediately after logging in.

### 6. Check which apps you have in Cloud

Any app that you are currently using with Jira Cloud application will need to be installed in your Jira application installation. For example, Gliffy, Tempo, etc.

Choose **Administration**  > **Manage apps**. The 'Find apps' screen shows apps available via the [Atlassian Marketplace](#). Choose **Manage apps** to view the apps currently installed on your Jira applications. Choose **Manage apps** and note the apps listed under the **User-installed Apps** section. You will need to note the app names and versions.

### 7. Install your app in Jira Server

For each app that you noted in the previous step, install it in your Jira applications. Make sure to install the latest version of the app. Atlassian does not provide support for data that is downgraded as a result of installing an older version of an app.

See [Managing Apps](#) for instructions on how to install an app. You will need to manually add the app license keys.

## 8. Install the Cloud compatibility for Jira app or run scripts

After you migrate from Cloud to Server, it might happen that user mentions are shown as AccountID rather than username. This is a known GDPR-related issue. To fix it, you can either use the Cloud compatibility for Jira app or run update scripts in your database.

The app is a quicker solution however it is a temporary one because it only fixes the UI side of things and replaces accountIDs from Jira Cloud with more user-friendly text in wiki markup fields. However, you can consider it if you want a quick fix and do not have time to test the scripts in your environment. This app is compatible with Jira Server 8.0 and later.

1. In your Jira Server instance go to **Jira Administration > Manage apps > Find new apps**.
2. Search for Cloud Compatibility for Jira.
3. Install the app.

To permanently remove the problem, run the update scripts in your database. Make sure that you run the scripts in your test environment first, and create a DB backup.

### PostgreSQL:

```
SELECT ('UPDATE jiraaction SET actionbody = replace(actionbody, 'accountid:' || cu.external_id || ' ',
' ' || cu.lower_user_name || ' ') WHERE actionbody LIKE '%[~accountid:' || cu.external_id || ']%'')
as "Queries to fix" FROM cwd_user cu WHERE cu.external_id IS NOT NULL;
```

### MySQL:

```
SELECT (CONCAT('UPDATE jiraaction SET actionbody = replace(actionbody, 'accountid:' , ifnull(cu.
external_id, ' ) , ' ', ' ', ifnull(cu.lower_user_name, ' ) , ' ') WHERE actionbody LIKE '%
[~accountid:' , ifnull(cu.external_id, ' ) , ']%'')) as "Queries to fix" FROM cwd_user cu WHERE cu.
external_id IS NOT NULL;
```

## 9. Reset the passwords for your users

After you finish migrating from Jira Cloud to Jira Server, you must reset the passwords for your users. To do this, you can either:

- notify all users that they need to click **Forgot password** to reset their password. [Learn how to send an email to all Jira users](#).
- log in as **sysadmin** and reset everyone's password.

**Well done! You've migrated your Jira Cloud site into a Jira Server instance.**

# Migrating from Jira Server to Cloud

When opened in a viewport, the user will be redirected to: [Jira server to cloud migration resources](#).

# Running Jira Data Center on a Kubernetes cluster

If you're running self-managed environments and looking to adopt modern infrastructures, Atlassian Data Center products can now be deployed on Kubernetes clusters. By leveraging Kubernetes, you can drive greater agility amongst your teams while experiencing a simplified administrative experience at scale, without compromising your organizations regulatory requirements.

We offer [Helm charts on GitHub](#) for installing and operating Atlassian products on a Kubernetes cluster of your choice. The Helm charts we offer are for:

- Jira
- Confluence
- Bitbucket
- Bamboo

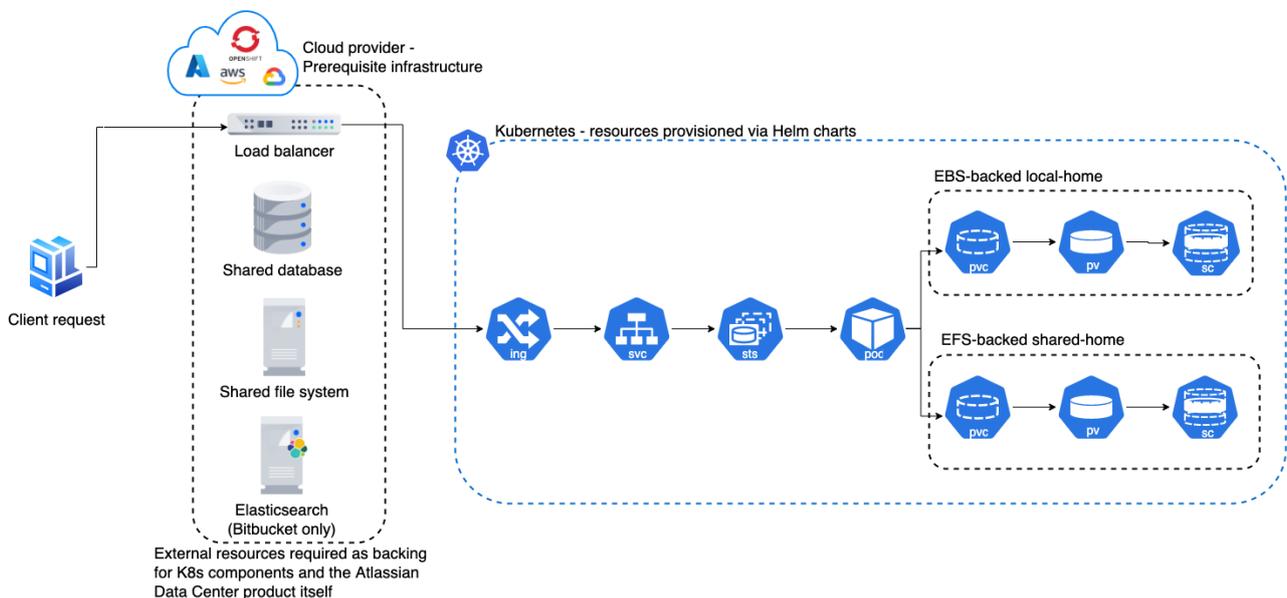
Helm is a package manager for Kubernetes that allows you to package, configure, and deploy applications and services onto Kubernetes clusters. Helm uses a packaging format called *charts*, which are collections of files that describe a related set of Kubernetes resources.

The Kubernetes cluster can be a managed environment, such as [Amazon EKS](#), [Azure Kubernetes Service](#), [Google Kubernetes Engine](#), or a custom on-premise system.

We strongly recommend you set up user management, central logging storage, a backup strategy, and monitoring, just as you would for a Data Center installation running on your own hardware.

## How it works

Here's an architectural overview of what you'll get when deploying your Data Center application on a Kubernetes cluster using the Helm charts:



Kubernetes entities required for product deployment:

- **Ingress and Ingress controller (ing)** - the Ingress defines the rules for traffic routing, which indicate where a request will go in the Kubernetes cluster. The Ingress controller is the component responsible for fulfilling those rules.
- **Service (svc)** - provides a single address for a set of pods to enable load-balancing between application nodes.
- **StatefulSets (sts)** - manages the deployment and scaling of a set of pods.

- **Pod** - a group of one or more containers, with shared storage and network resources, and a specification for how to run the containers. Pods are the smallest deployable units of computing that you can create and manage in Kubernetes.
- **PersistentVolumeClaim** (pvc) - reserves the Persistent Volume (PV) to be used by a pod.
- **PersistentVolume** (pv) - is the "physical" volume on the host machine that stores your persistent data.
- **StorageClass** (sc) - provides a way for administrators to describe the "classes" of storage they offer.

## Installing your Data Center application on a Kubernetes cluster

In addition to the Helm charts we provide for installing the Atlassian Data Center applications, we also provide [documentation and examples on a dedicated documentation site](#).

Before you begin, you need to have an understanding of [Kubernetes](#) and [Helm](#) concepts.

To install and operate your Data Center application on a Kubernetes cluster using our Helm charts:

1. Follow the requirements and set up your environment according to the [Prerequisites guide](#).
2. Perform the installation steps described in the [Installation guide](#).
3. Learn how to upgrade applications, scale your cluster, and update resources using the [Operation guide](#).

# Getting started with Jira Data Center on AWS

If you decide to deploy Jira Data Center in a clustered environment, consider using Amazon Web Services (AWS). AWS allows you to scale your deployment elastically by resizing and quickly launching additional nodes, and provides a number of managed services that work out of the box with Jira Data Center. These services make it easier to configure, manage, and maintain your deployment's clustered infrastructure.

Interested in learning more about what Jira Data Center provides? [Click here](#) for an overview.

## On this page:

- [Non-clustered VS clustered environment](#)
- [Deploying Jira Data Center in a cluster using the AWS Quick Start](#)
  - [Advanced customizations](#)
  - [Launching the Quick Start from your own S3 bucket \(recommended\)](#)
  - [Amazon Aurora database for high availability](#)
  - [Amazon CloudWatch for basic monitoring and centralized logging](#)
  - [Auto Scaling groups](#)
- [Supported AWS regions](#)
- [Customizing the AWS Quick Start's CloudFormation templates](#)

## Non-clustered VS clustered environment

A single node is adequate for most Small or Medium size deployments, unless you need high availability or [zero-downtime upgrades](#).

If you have an existing Server installation, you can still use its infrastructure when you upgrade to Data Center. Many features exclusive to Data Center (like [SAML single sign-on](#), [self-protection via rate limiting](#), and [CDN support](#)) don't require clustered infrastructure. You can start using these Data Center features by simply upgrading your Server installations license. For more information on whether clustering is right for you, check out [Atlassian Data Center architecture and infrastructure options](#).

## Deploying Jira Data Center in a cluster using the AWS Quick Start

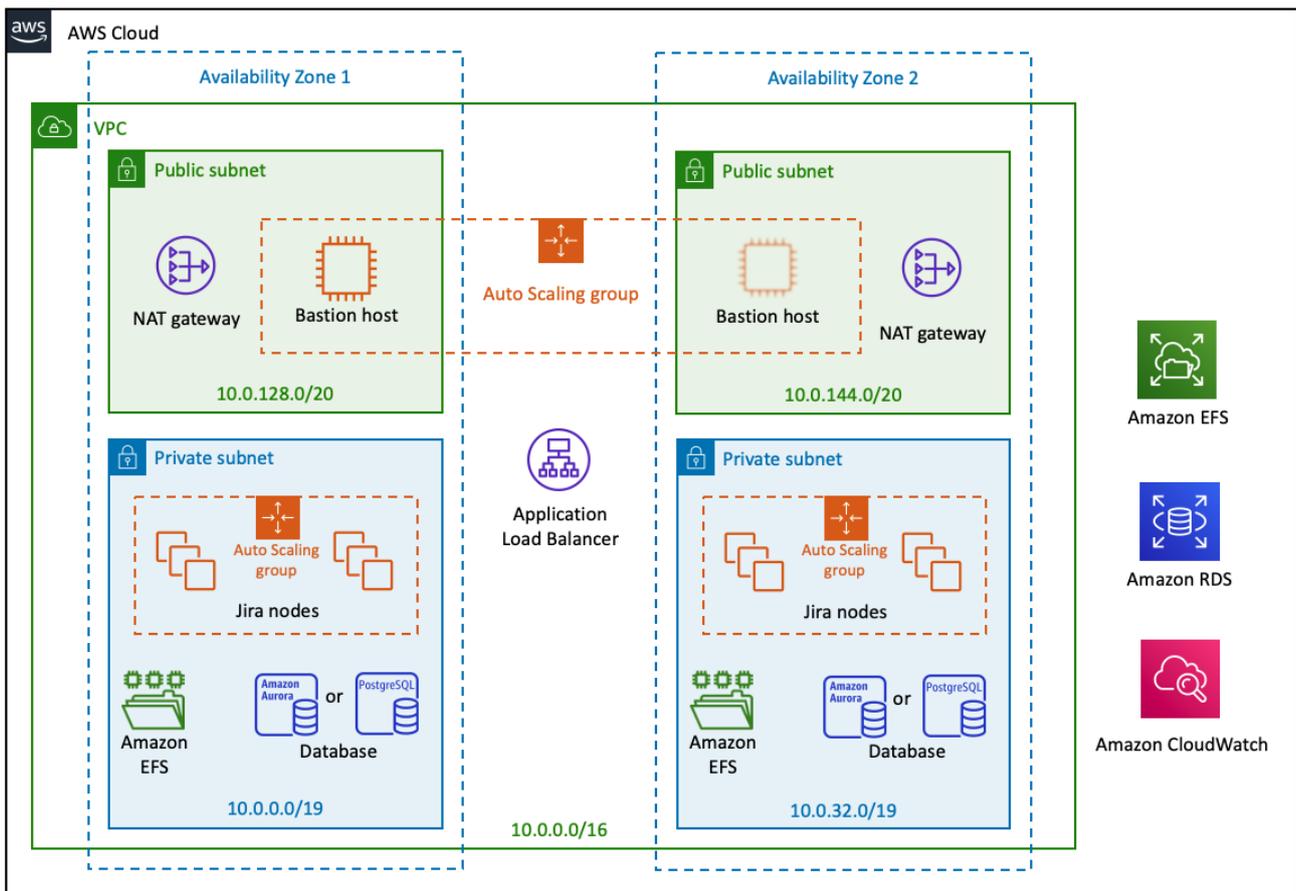
The simplest way to deploy your entire Data Center cluster in AWS is by using the Quick Start. The Quick Start launches, configures, and runs the AWS compute, network, storage, and other services required to deploy a specific workload on AWS, using AWS best practices for security and availability.

The Quick Start provides two deployment options, each with its own template. The first option deploys the Atlassian Standard Infrastructure (ASI) and then provisions either Jira Software or Jira Service Desk Data Center into this ASI. The second option only provisions Jira Software or Jira Service Desk Data Center on an existing ASI.

### **Atlassian Standard Infrastructure (ASI)**

The ASI is a virtual private cloud (VPC) that contains the components required by all Atlassian Data Center applications. For more information, see [Atlassian Standard Infrastructure \(ASI\) on AWS](#).

Here's an overview of the architecture for the Jira Data Center Quick Start:



The deployment consists of the following components:

- **Instances/nodes:** One or more Amazon Elastic Cloud (EC2) instances as cluster nodes, running Jira.
- **Load balancer:** An Application Load Balancer (ALB), which works both as load balancer and SSL-terminating reverse proxy.
- **Amazon EFS:** A shared file system for storing artifacts in a common location, accessible to multiple Jira nodes. The Quick Start architecture implements the shared file system using the highly available Amazon Elastic File System (Amazon EFS) service.
- **Database:** Your choice of shared database instance Amazon RDS or Amazon Aurora.
- **Amazon CloudWatch:** Basic monitoring and centralized logging through Amazon's native CloudWatch service.

For more information, see [Jira products on AWS](#).

### Advanced customizations

To get you up and running as quickly as possible, the Quick Start doesn't allow the same level of customization as a manual installation. You can, however, further customize your deployment through the variables in the [Ansible playbooks](#) we use.

All of our AWS Quick Starts use Ansible playbooks to configure specific components of your deployment. These playbooks are available publicly on this repository:

<https://bitbucket.org/atlassian/dc-deployments-automation>

You can override these configurations by using [Ansible variables](#). Refer to the repository's [README file](#) for more information.

### ✔ jira-config.properties customizations

Jira Server allows you to apply advanced settings through the `jira-config.properties` file. You can also use the same file to apply the same settings to an existing Quick Start deployment. For detailed instructions on how to do this, see [Use the jira-config.properties file to customize an AWS Quick Start deployment](#).

## Launching the Quick Start from your own S3 bucket (recommended)

The fastest way to launch the Quick Start is directly from its AWS S3 bucket. However, when you do, any updates we make to the Quick Start templates will propagate directly to your deployment. These updates sometimes involve adding or removing parameters from the templates. This could introduce unexpected (and possibly breaking) changes to your deployment.

For production environments, we recommend that you copy the Quick Start templates into your own S3 bucket. Then, launch them directly from there. Doing this gives you control over when to propagate Quick Start updates to your deployment.

1. Clone the [Quick Start templates](#) (including all of its submodules) to your local machine. From the command line, run:

```
git clone --recurse-submodules https://github.com/aws-quickstart/quickstart-atlassian-jira.git
```

2. **(Optional)** The [Quick Start templates](#) repository uses the directory structure required by the Quick Start interface. If needed (for example, to minimize storage costs), you can remove all other files except the following:

```
quickstart-atlassian-jira
submodules
  quickstart-atlassian-services
  templates
  quickstart-vpc-for-atlassian-services.yaml
  templates
  quickstart-jira-dc-with-vpc.template.yaml
  quickstart-jira-dc.template.yaml
```

3. Install and set up the [AWS Command Line Interface](#). This tool will allow you to create an S3 bucket and upload content to it.
4. Create an S3 bucket in your region:

```
aws s3 mb s3://<bucket-name> --region <AWS_REGION>
```

At this point, you can now upload the Quick Start templates to your own S3 bucket. Before you do, you'll have to choose which Quick Start template you'll be using:

- `quickstart-jira-dc-with-vpc.template.yaml`: use this for deploying into a new ASI (end-to-end deployment).
- `quickstart-jira-dc.template.yaml`: use this for deploying into an existing ASI.

1. In the template you've chosen, the `QSS3BucketName` default value is set to `aws-quickstart`. Replace this default with the name of your S3 bucket.

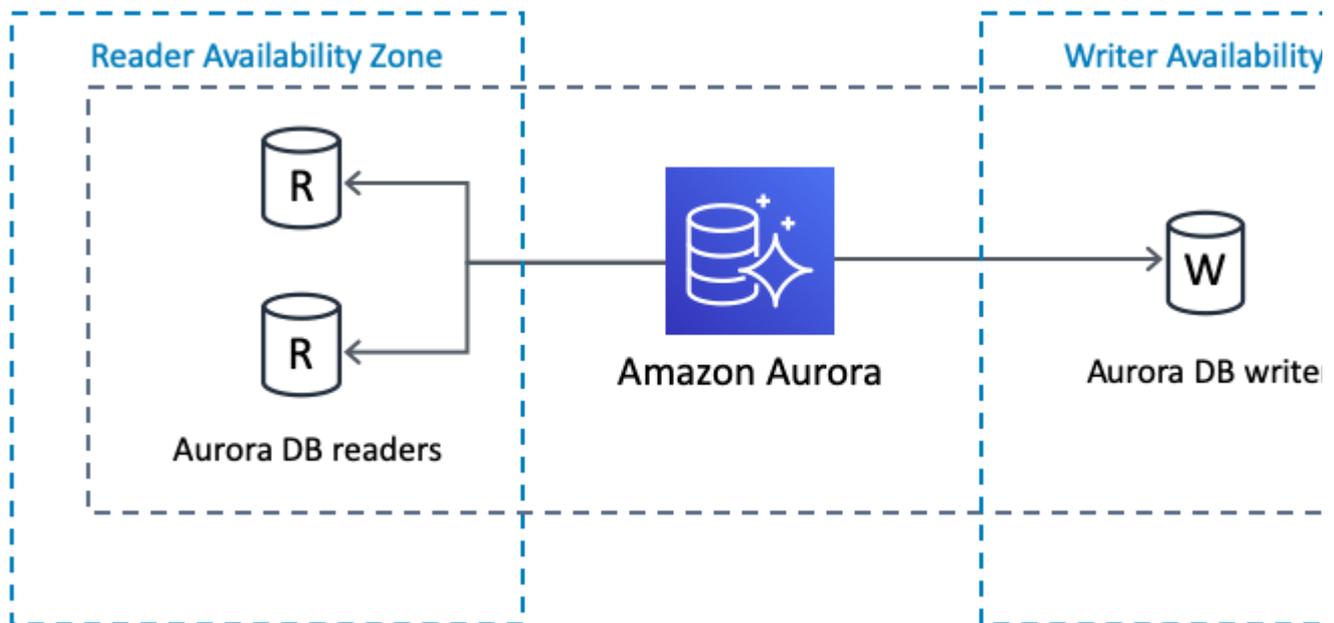
- Go into the parent directory of your local clone of the Quick Start templates. From there, upload all the files in local clone to your S3 bucket:

```
aws s3 cp quickstart-atlassian-jira s3://<bucket-name> --recursive --acl public-read
```

- Once you've uploaded everything, you're ready to deploy your production stack from your S3 bucket. Go to **CloudFormation Create Stack**. When specifying a template, paste in the Object URL of the Quick Start template you'll be using.

### Amazon Aurora database for high availability

The Quick Start also allows you to deploy Jira Data Center with an Amazon Aurora clustered database (instead of RDS). This cluster will be PostgreSQL-compatible, featuring a primary database writer that replicates to two database readers. You can also set up the writers and readers in separate availability zones for better resiliency.



If the writer fails, Aurora automatically promotes one of the readers to take its place. For more information, see [Amazon Aurora Features: PostgreSQL-Compatible Edition](#).

**i** For instructions on manually setting up a new Amazon Aurora clustered database and connecting it to Jira Data Center, see [Connecting Jira Data Center to Amazon Aurora](#). Amazon Aurora is supported on [Jira Software 8.4](#), [Jira Service Desk 4.4](#), and all later versions of both.

### Amazon CloudWatch for basic monitoring and centralized logging

The Quick Start can also provide you with node monitoring through [Amazon CloudWatch](#). This will allow you to track each node's CPU, disk, and network activity, all from a pre-configured [CloudWatch dashboard](#). The dashboard will be configured to display the latest log output, and you can customize the dashboard later on with additional monitoring and metrics.

By default, Amazon CloudWatch will also collect and store logs from each node into a single, central source. This centralized logging allows you to search and analyze your deployment's log data more easily and effectively. See [Analyzing Log Data with CloudWatch Logs Insights](#) and [Search Log Data Using Filter Patterns](#) for more information.

**i** Amazon CloudWatch provides basic logging and monitoring, but also costs extra. To help reduce the cost of your deployment, you can disable logging or turn off Amazon CloudWatch integration during deployment.

✔ To download your log data (for example, to archive it or analyze it outside of AWS), you'll have to export it first to S3. From there, you can download it. See [Exporting Log Data to Amazon S3](#) for details.

## Auto Scaling groups

This Quick Start uses [Auto Scaling groups](#), but only to statically control the number of its cluster nodes. We don't recommend that you use Auto Scaling to dynamically scale the size of your cluster. Adding an application node to the cluster usually takes more than 20 minutes, which isn't fast enough to address sudden load spikes.

If you can identify any periods of high and low load, you can schedule the application node cluster to scale accordingly. See [Scheduled Scaling for Amazon EC2 Auto Scaling](#) for more information.

To study trends in your organization's load, you'll need to monitor the performance of your deployment. Refer to [Jira Data Center sample deployment and monitoring strategy](#) for tips on how to do so.

## Supported AWS regions

Not all regions offer the services required to run Jira. You'll need to choose a region that supports Amazon Elastic File System (EFS). These regions are:

- Americas
  - Northern Virginia
  - Ohio
  - Oregon
  - Northern California
  - Montreal
- Europe/Middle East/Africa
  - Ireland
  - Frankfurt
  - London
  - Paris
- Asia Pacific
  - Singapore
  - Tokyo
  - Sydney
  - Seoul
  - Mumbai

This list was last updated on 20 Jun 2019.

The services offered in each region change from time to time. If your preferred region isn't on this list, check the [Regional Product Services](#) table in the AWS documentation to see if it already supports EFS.

### **AWS GovCloud is not for production**

Our Quick Starts are also available on [AWS GovCloud](#) regions, but only for testing purposes. We do not support any deployments on GovCloud regions resulting from this Quick Start.

## Customizing the AWS Quick Start's CloudFormation templates

To get you up and running as quickly as possible, the Quick Start doesn't allow the same level of customization as a manual installation. Alternatively, you can customize the CloudFormation templates used by the Quick Start to fit your needs. These templates are available from the following repository:

<https://github.com/aws-quickstart/quickstart-atlassian-jira>

### Related pages:

- [Administering Jira Data Center on AWS](#)
- [Upgrading Jira Data Center in AWS](#)

# Administering Jira Data Center on AWS

- [Setting custom DNS name](#)
- [Scaling up and down](#)
- [Connecting to your nodes over SSH](#)
- [Backing up](#)
- [Migrating your existing instance into AWS](#)

While working with your Jira Data Center on AWS, you can expand your environment by adding additional nodes, upgrade the existing Jira instances, or connect to them over SSH.

## Setting custom DNS name

When deploying Jira Data Center on AWS, you get a default domain name that points to the Amazon's load balancer. You'll be using it to access Jira. This domain name depends on the load balancer's name and the AWS region, but in general it looks like this: `my-loadbalancer-1234567890.us-west-2.elb.amazonaws.com`. You can change it to something more familiar, e.g. `jira.atlassian.com`, by entering your own domain name in the **Existing DNS (optional)** parameter in the Quick Start. You'll need a domain name to do this, if you don't have one already, you can register it [here](#).

To set the custom DNS name:

1. When deploying Jira with the Quick Start, enter your domain name (FQDN) in the **Existing DNS (optional)** parameter. It'll be saved in the `proxyName` parameter in Apache Tomcat, which is a web server used by Jira. All nodes will be using this domain name.
2. After the deployment, when you know the address of the Amazon's load balancer, associate it with your domain name. To do this, you'll need to use your DNS service to create a CNAME record where you enter the source and target URLs, creating an alias. See [Associating your custom domain name with your load balancer name](#).

**If you have already deployed Jira**, you can still change the parameters that are used by your stack, be it the instance type or the domain name. See [Changing resource properties](#).

## Scaling up and down

To change the number of nodes in the cluster:

1. Sign in to the AWS Management Console, use the region selector in the navigation bar to choose the AWS Region for your deployment, and open the AWS CloudFormation console at <https://console.aws.amazon.com/cloudformation/>.
2. Click the **Stack name** of your deployment. This will display your deployment's **Stack info**. From there, click **Update**.
3. On the **Select Template** page, leave **Use current templates** selected, and then choose **Next**.
4. On the **Specify Details** page, go to the **Cluster nodes** section of **Parameters**. From there, set your desired number of application nodes in the following parameters:
  - a. **Minimum number of cluster nodes**
  - b. **Maximum number of cluster nodes**
5. Click through to update the stack.

### Disabled Auto Scaling

Since your cluster has the same minimum and maximum number of nodes, [Auto Scaling](#) is effectively disabled.

Setting different values for the minimum and maximum number of cluster nodes enables Auto Scaling. This dynamically scale the size of your cluster based on system load.

However, we recommend that you keep Auto Scaling disabled. At present, Auto Scaling can't effectively address sudden spikes in your deployment's system load. This means that you'll have to manually re-scale your cluster depending on the load.

## Connecting to your nodes over SSH

You can perform node-level configuration or maintenance tasks on your deployment through the [AWS Systems Manager Sessions Manager](#). This browser-based terminal lets you access your nodes without any SSH Keys or a Bastion host. For more information, see [Getting started with Session Manager](#).

### Access via Bastion host

You can also access your nodes via a Bastion host (if you deployed one). To do this, you'll need your SSH private key file (the PEM file you specified for the **Key Name** parameter). Remember, this key can access all nodes in your deployment, so keep this key in a safe place.

The Bastion host acts as your "jump box" to any instance in your deployment's internal subnets. That is, access the Bastion host first, and from there access any instance in your deployment.

The Bastion host's public IP is the `BastionPubIp` output of your deployment's `ATL-BastionStacks` tack. This stack is nested in your deployment's [Atlassian Standard Infrastructure \(ASI\)](#). To access the Bastion host, use `ec2-user` as the user name, for example:

```
ssh -i keyfile.pem ec2-user@<BastionPubIp>
```

The `ec2-user` has `sudo` access. SSH access is by `root` is not allowed.

## Backing up

We recommend you use the AWS native backup facility, which utilizes snapshots to back up your Jira Data Center. For more information, see [AWS Backup](#).

## Migrating your existing instance into AWS

To migrate an existing instance into AWS:

1. Migrate its database to PostgreSQL (if not already).
2. Take a backup of your existing home directory and database.
3. Copy the backup file to your file server EC2 instance.
4. Unpack the backup file under `/media/at1/jira/shared` of your file server.
5. Restore the PostgreSQL database dump contained in the backup file to your RDS instance with `pg_restore`.

# Upgrading Jira Data Center on AWS

- [Choosing an upgrade method](#)
- [Option 1: Normal upgrade](#)
  - [Step 1: Terminate all running Jira Data Center application nodes](#)
  - [Step 2: Update the version used by your Jira Data Center stack](#)
  - [Step 3: Scale up the number of application nodes](#)
- [Option 2: Zero downtime upgrade](#)
  - [Step 1: Put Jira into upgrade mode](#)
  - [Step 2: Find all the current application nodes in your stack](#)
  - [Step 3: Update the version used by your Jira Data Center stack](#)
  - [Step 4: Terminate all nodes running the older version](#)
  - [Step 5: Finalize the upgrade](#)

Before you begin, consider upgrading to an [Atlassian Long Term Support releases](#) (if you're not on one already). Long Term Support releases get fixes for critical bugs and security issues throughout its two-year support window. This gives you the option to keep a slower upgrade cadence without sacrificing security or stability. Long Term Support releases are suitable for companies who can't keep up with the frequency at which we ship feature releases.

Here's some useful advice for upgrading your deployment:

1. Before upgrading to a later version of Jira Data Center, [check if your apps are compatible](#) with that version. [Update your apps](#) if needed. For more information about managing apps, see [Using the Universal Plugin Manager](#).
2. If you need to keep Jira Data Center available to users during your upgrade, we recommend [upgrading Jira Data Center with zero downtime](#). This method allows your nodes to work on different Jira versions while you upgrade them one by one. During the upgrade, Jira remains available to users.
3. We strongly recommend that you perform the upgrade first in a [staging environment](#) before upgrading your production instance. [Creating a test environment for Jira](#) provides helpful tips on doing so.

## Choosing an upgrade method

If you deployed using our [AWS Quick Start](#), you can upgrade your Jira version using either method:

- [Normal upgrade](#): this process is similar to upgrading [Confluence Data Center](#) or [Bitbucket Data Center](#), and will involve some downtime.
- [Zero downtime upgrade](#): this process involves more steps than a normal upgrade, but will not involve any downtime.

## Option 1: Normal upgrade

Performing a normal Jira Data Center upgrade on AWS involves three steps:

### Step 1: Terminate all running Jira Data Center application nodes

Set the number of application nodes used by the Jira Data Center stack to 0. Then, update the stack. Doing this will make Jira Data Center unavailable (until you finish the next step).

1. In the AWS console, go to **Services > CloudFormation**. Select your deployments stack to view its Stack Details.
2. In the Stack Details screen, click **Update Stack**.
3. From the **Select Template** screen, select **Use current template** and click **Next**.
4. You'll need to terminate all running nodes. To do that, set the following parameters to 0:
  - a. **Maximum number of cluster nodes**
  - b. **Minimum number of cluster nodes**
5. Click **Next**. Click through the next pages, and then to apply the change using the **Update** button.
6. Once the update is complete, check that all application nodes have been terminated.

## Step 2: Update the version used by your Jira Data Center stack

Set the number of application nodes used by Jira Data Center to 1. Configure it to use the version you want. Then, update the stack again.

1. From your deployments Stack Details screen, click **Update Stack** again.
2. From the **Select Templates** screen, select **Use current template** and click **Next**.
3. Set the **Version** parameter to the version you're updating to.
4. Configure your stack to use one node. To do that, set the following parameters to 1:
  - a. **Maximum number of cluster nodes**
  - b. **Minimum number of cluster nodes**
5. Click **Next**. Click through the next pages, and then to apply the change using the **Update** button.

## Step 3: Scale up the number of application nodes

You can now scale up your deployment to your original number of application nodes. For detailed instructions on how to do this, see [Scaling up and down](#).

### Option 2: Zero downtime upgrade

Performing a zero downtime upgrade of Jira Data Center on AWS involves four steps:

#### Step 1: Put Jira into upgrade mode

Put Jira into upgrade mode to allow your nodes to work on different versions while you upgrade them one by one.

1. In Jira, go to  **Applications > Jira upgrades**.
2. Click **Put Jira into upgrade mode**. This will only be available if your nodes are all on the same version.

#### Step 2: Find all the current application nodes in your stack

In AWS, note the Instance IDs of all running application nodes in your stack. These are all the application nodes running the older version of Jira. You'll need these IDs for a later step.

1. In the AWS console, go to **Services > CloudFormation**. Select your deployments stack to view its Stack Details.
2. Expand the **Resources** drop-down. Look for the **ClusterNodeGroup** and click its Physical ID. This will take you to a page showing the Auto Scaling Group details of your application nodes.
3. In the Auto Scaling Group details, click on the **Instances** tab. Note all of the Instance IDs listed there; you'll be terminating them at a later step.

#### Step 3: Update the version used by your Jira Data Center stack

Increase the number of application nodes used by Jira Data Center by 1. Configure it to use the version you want. Then, update the stack.

1. From your deployments Stack Details screen, click **Update Stack**.
2. From the **Select Templates** screen, select **Use current template** and click **Next**.
3. Set the **Version** parameter to the version you're updating to.
4. Increase the number of Jira application nodes used by your stack by 1. To do that, increase the following parameters:

- a. **Maximum number of cluster nodes**
  - b. **Minimum number of cluster nodes**
5. Click **Next**. Click through the next pages, and then to apply the change using the **Update** button.

#### Step 4: Terminate all nodes running the older version

In [Step 2](#), you noted all nodes running the older version of Jira. You'll need to terminate them one by one.

To terminate a node or instance:

1. In the AWS console, go to **Services > EC2**. From there, click **Running Instances**.
2. Check one of the instances matching the Instance IDs you noted in [Step 2](#).
3. From the **Actions** drop-down, select **Instance State > Terminate**.
4. Click through to terminate the instance.

Each time you terminate a node, AWS will automatically replace it. The new replacement node will be running the new, upgraded Jira version. Upon terminating a node, wait for its replacement to boot fully and join the cluster before terminating another node. This will help ensure that your cluster has enough nodes to handle user load.

#### Step 5: Finalize the upgrade

This step will allow any required upgrade tasks to run on your instance, and take Jira out of upgrade mode. Once the required tasks have completed, Jira will be fully upgraded.

1. In Jira, go to  **> Applications > Jira upgrades**.
2. Click **Finalize upgrade**. This will only be available if all your nodes are all on the same, new version.

Well done! You upgraded your Jira instance to a new version. You can see the current version by going to  **> Applications > Versions and licenses**.

# Getting started with Jira Data Center on Azure

If you decide to deploy Jira Data Center in a clustered environment, consider using Microsoft Azure. This platform allows you to scale your deployment elastically by resizing and quickly launching additional nodes and provides a number of managed services that work out of the box with Jira Data Center. These services make it easier to configure, manage, and maintain your deployment's clustered infrastructure.

We've set up reference templates that let you deploy [Jira Software Data Center](#) and [Jira Service Desk Data Center](#) in Microsoft Azure, and you can then configure it depending on your organization's needs. We strongly recommend you to set up user management, central logging storage, a backup strategy, and monitoring, just as you would for a Jira Data Center installation running on your own hardware.

## On this page:

[Non-clustered VS clustered environment](#)

[How it works](#)

[Standardized infrastructure](#)

[Migrating to an Azure deployment](#)

[Before you begin](#)

[Steps](#)

[Securing your Azure deployment](#)

[Preparing for your deployment](#)

[1. Determining the size of your deployment](#)

[2. Choosing the region](#)

[3. Preparing additional information](#)

[Deploying Jira Data Center to Azure via Azure Marketplace](#)

[Jira-specific parameters](#)

[Standardized infrastructure parameters](#)

[Deploying Jira Data Center to Azure using the CLI](#)

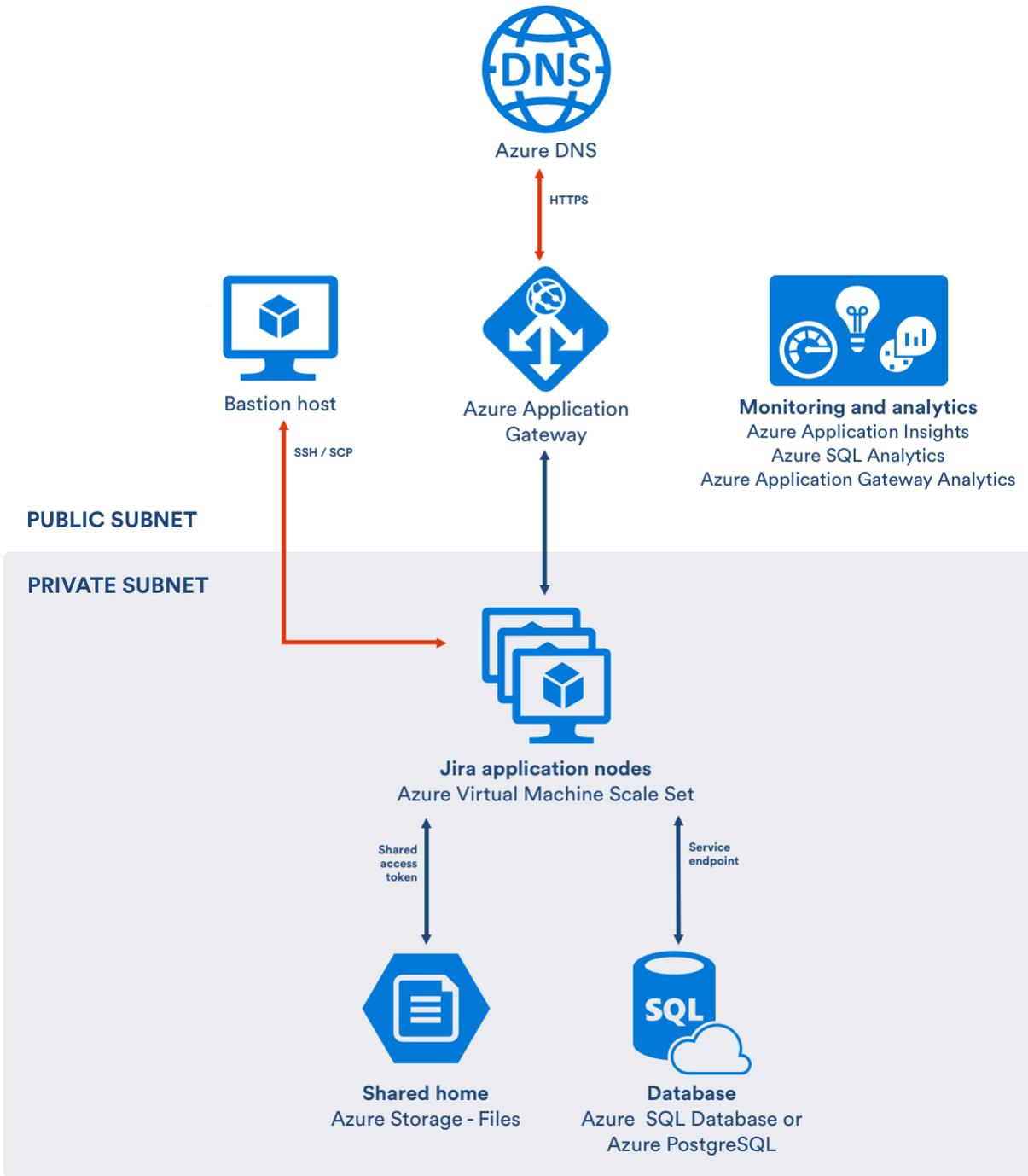
[Monitoring](#)

## Non-clustered VS clustered environment

A single node is adequate for most small or medium size deployments, unless you need high availability or [zero-downtime upgrades](#). If you have an existing Server installation, you can still use its infrastructure when you upgrade to Data Center. Many features exclusive to Data Center (like [SAML single sign-on](#), [self-protection via rate limiting](#), and [CDN support](#)) don't require clustered infrastructure. You can start using these Data Center features by simply upgrading your Server installations license. For more information on whether clustering is right for you, check out [Atlassian Data Center architecture and infrastructure options](#).

## How it works

Here's an architectural overview of what you'll get when deploying Jira Data Center using the templates:



The deployment contains one or more Azure standard VM instances as application cluster nodes in a scale set. Each node runs Jira Software or Jira Service Desk, depending on which template you use.

The template also provisions an Azure Files storage account for the shared home. This shared home stores attachments and other files accessible to the application cluster nodes. It's mounted as a SAN drive on each cluster node and treated normally like any other file.

### Standardized infrastructure

The [Jira Data Center](#), [Confluence Data Center](#), [Bitbucket Data Center](#), and [Crowd Data Center](#) templates deploy the following infrastructure components identically:

Component	Configuration
Bastion host	This is a lightweight but highly secure Azure Linux VM that controls SSH access to the application cluster nodes.

Application Gateway	By default, this gateway is composed of two instances for high availability. It acts as a HTTP/HTTPS load balancer for your scale set of application cluster nodes.
Monitoring	The ARM templates configure Azure Monitoring to perform basic health and availability monitoring to cluster nodes and database.
Database	You can choose between Azure SQL Database (MS SQL Server-compatible) or Azure PostgreSQL database. Either way, the database will be configured as service endpoints to only allow traffic from the private network that the cluster nodes are in. This restricted traffic setup helps enhance security.

## Migrating to an Azure deployment

You can also migrate your existing Jira Data Center instance into Azure. To do this, you will need to set up a new Jira Data Center instance in Azure, and then import data from your existing instance. This approach ensures that your new site is created with optimum settings for Azure.

Here's a high-level overview of the steps.

### Before you begin

Back up your existing Jira instance, including your database and home directories.

### Steps

1. Make a list of any Marketplace or other user-installed apps.
2. [Export](#) the data into XML format. The attachments won't be exported, so you need to manually copy them from your current shared home directory.
3. Deploy Jira Data Center in Azure, and test that Jira is working as expected.
4. [Import](#) the XML backup that you created before.
5. Copy the contents of your /attachments directory to the equivalent directory in your shared home directory in Azure.
6. Install any apps that you used before.
7. Test your new Jira Data Center instance.

## Securing your Azure deployment

We recommend deploying Jira with SSL. Our template will prompt you for a certificate and password.

Good to know:

- HTTPS is terminated at the application gateway.
- Your certificate should be from a trusted Certificate Authority. You should avoid self-signed certificates.

## Preparing for your deployment

### 1. Determining the size of your deployment

While deploying Jira Data Center on Microsoft Azure, you'll have an option to choose the size of your deployment: small, medium, large, or enterprise. This will determine the number of Jira application nodes, and the size of VMs to be provisioned.

To help you estimate your instances size, we identified several Jira Data Center size profiles that are based on the amount of data dimensions that you already have, or are planning to have, such as the number of issues, projects, custom fields, users, and so on. For more information, see [Jira Data Center size profiles](#).

### 2. Choosing the region

The region, or location, is where Azure will house your deployment. Some regions don't provide all Azure features, like access to Application Insights and analytics. The choice here depends on what features would you like to use for your deployment.

To browse regions and available features, see [Products available by region](#).

### 3. Preparing additional information

During the deployment, you will also need:

- **(Optional) SSL:** A Base64 encoded PFX certificate from a trusted Certificate Authority.
- **(Optional) CNAME:** Details of your existing CNAME, if you don't want Azure to generate a random domain for you.

### Deploying Jira Data Center to Azure via Azure Marketplace

This method uses the Azure Marketplace to deploy Jira Software Data Center or Jira Service Desk Data Center using our deployment templates as a reference.

To deploy Jira Data Center to Azure using our Marketplace app:

1. Sign in to [Azure Portal](#).
2. Select **Create a resource** to start a new deployment
3. Search for **Atlassian**, then select [Jira Software Data Center](#) or [Jira Service Desk Data Center](#) from the list of Marketplace apps.
4. Select **Create** to start configuring the deployment.
5. Follow the prompts in the wizard to configure your deployment. Refer to the following parameters table for more information.
6. Confirm all the details are correct, then select **Create** to purchase the subscription. Deployment will take about 30 minutes.
7. Once deployment is complete, go to the Jira URL listed in the deployment outputs to complete onboarding and start using Jira.

### Jira-specific parameters

The following parameters apply only to Jira Software and Jira Service Desk.

<b>Jira version</b>	Specify the version of Jira you'd like to install in full. For example, 7.13.0.  We'd recommend that you choose Jira Software 7.13 or later or Jira Service Desk 3.16 or later. These versions have been thoroughly tested in Azure. Head to <a href="#">Jira Software Release Notes</a> or <a href="#">Jira Service Desk Release Notes</a> for a list of all releases.
<b>Jira admin credentials</b>	Provide a name and password for the initial Jira administrator on your instance.
<b>Jira cluster</b>	Select the expected size of your site trial, small, medium, large, or enterprise. This will determine the number of Jira application nodes and the size of VMs to be provisioned. Select <b>Change Size</b> to override the defaults.

### Standardized infrastructure parameters

The [Jira Data Center](#), [Confluence Data Center](#), [Bitbucket Data Center](#), and [Crowd Data Center](#) templates all share the same parameters:

Parameter	Description
Subscription	Your Microsoft Azure subscription type.

Resource group	If you have an existing resource group, you can use it, or create a new one.
Location	This is the region where Azure will house your deployment.
SSH Access	Provide an SSH public key to be used to SSH into the instance that will act as bastion host, and a username and password for SSH access to the Bitbucket nodes. See <a href="#">Create and use an SSH public-private key pair for Linux VMs in Azure</a> in the Microsoft Azure documentation.
Database configuration	Choose between an Azure SQL Database, or Azure Database for PostgreSQL. Provide a username and password for the database admin user.  <div style="border: 1px solid #ffc107; padding: 10px; margin: 10px 0;"> <p> <b>Existing databases</b></p> <p>If you want to integrate with an existing database, you'll have to <a href="#">deploy to Azure using the CLI</a>.</p> </div>
CNAME	This is the <a href="#">Canonical Name record</a> (CNAME) for your organization. If you don't provide one, Azure will generate a random sub domain for your instance.
HTTP/SSL	Provide the certificate and password to be used for SSL termination on the Azure Application Gateway.
Monitoring	Choose the monitoring and analytics services that you would like to enable. Subject to availability in your location. See <a href="#">Monitoring</a> for related information.

## Deploying Jira Data Center to Azure using the CLI

This method uses the Azure command-line interface to deploy Jira Data Center using our deployment templates as a reference. You'll need to [install the Azure CLI](#) to do this.

Using the deployment templates directly allows for greater configuration granularity. All hardware choices such as the number of cluster nodes, size, disk size, and OS type are configurable as parameters.

Head to [atlassian-azure-deployment](#) in Bitbucket and check out the README to find out how to deploy using the CLI.

## Monitoring

As a number of the resources we provision are managed by Azure, a number of options are available for monitoring. For example:

- The application gateway will automatically monitor its backend pool (the Jira application nodes), sending the alerts to the Jira admin email address specified in the deployment. See [Application Gateway health monitoring overview](#) in the Microsoft Azure documentation.
- Application Insights can be used to see the overall system health and dig into particular areas of interest [Application Insights](#) in the Azure documentation.
- Azure SQL Analytics is available for more granular monitoring of your SQL Server database. [Monitor Azure SQL Database using Azure SQL Analytics](#) in the Microsoft Azure documentation.

Note that some of these resources are still in preview, so may not be available in your region yet.

# Administering Jira Software Data Center on Azure

Once you've deployed Jira Software Data Center to Azure using the reference deployment template, administering the application is similar to managing an application on your own hardware, with the exception that you'll need to go via the jumpbox to access your nodes and shared home directory.

## Connecting to your Azure jumpbox and nodes over SSH

Your cluster nodes have been deployed into a private subnet, which means you can't access them directly. To work around that, we've also deployed a small Azure VM (jumpbox, or bastion host) into a public subnet, so you can use it to access your nodes over SSH.

### Before you begin

- Get the SSH credentials you provided during the deployment.
- Get the private key file that you created for SSH during the deployment.
- Get the IP address of your jumpbox. In Azure Portal, open your resource group, and then go to **Deployments > atlassian.JIRA-data-center-*<id>* > Outputs**.
- Get the private IP address of the instance you want to access. In your resource group, open the **JIRA Cluster** resource, and go to **Instances**.

1. Access the jumpbox through a command line:

```
ssh -i privatekeyfile ssh_username@dns_name_or_ip_address
```

For example:

```
ssh -i privatekey JIRAadmin@JIRA-jumpbox-ip-73kaq.eastus.cloudapp.azure.com
```

2. Once you've accessed the jumpbox, you can connect to any of your nodes in the cluster:

```
ssh ssh_username@node_ip_address
```

For example:

```
ssh JIRAadmin@10.0.2.4
```

## Accessing your `dbconfig.xml` or `server.xml` files

Once you've accessed your nodes over SSH, navigate to the following directories to find your `server.xml` and `dbconfig.xml` files.

- **server.xml**: `/media/atl/JIRA/shared/server.xml`
- **dbconfig.xml**: `/media/atl/JIRA/shared/dbconfig.xml`

In Azure, these files are copied from the *shared home* to the *local home* when a **new** node joins the cluster.

When making changes to these files on existing nodes, it's important to also update the files in the shared home, otherwise a new node joining the cluster will be set up with outdated configuration.

So, when making changes in `server.xml` or `dbconfig.xml` on existing nodes, it's important to also update the files in the *shared home*, otherwise a new node joining the cluster will be set up with outdated configuration.

These files are only accessible from the existing nodes. The *shared home* is mounted (think of it as a network hard disk) on each node under `/media/atl/JIRA/shared`.

## Backing up and recovering from failures

We recommend you use the Azure native backup facilities where possible to make sure your data is backed up, and you can easily recover in the case of a failure.

## Database

We use Azure managed database instances with high availability. Azure provides several excellent options for backing up your database, so you should take some time to work out which will be the best, and most cost effective option for your needs. See the following Azure documentation for your chosen database:

- [SQL Database: Automated backups](#)
- [SQL Database: Backup retention](#)
- [PostgreSQL: Backup concepts](#)

## Shared home

The shared home is where your attachments and export files are stored. We create a general purpose Azure storage account, configured with [local redundant storage](#), which means there are multiple copies of the data at any one time.

Because of the redundancy strategy, it shouldn't be necessary to take regular backups, however you can take point in time backups using [snapshots](#) if necessary.

## Application nodes

The application nodes are VMs in an [Azure Virtual Machine Scale Set](#). Each application node has a JIRA installation directory and a local home directory containing things like logs and search indexes.

Like the shared home, application nodes are configured with [local redundant storage](#), which means there are multiple copies of the data at any one time.

If you've manually customised any configuration files in the installation directory (for example velocity templates), you may also want to manually back these up as a reference.

## Bastion host

As this VM acts as a jumpbox, and doesn't store any data it doesn't need to be backed up. If the VM becomes unresponsive it can be restarted from the Azure Portal.

## Application gateway

The application gateway is highly available. We deploy 2 instances by default. As with the bastion host, it doesn't need to be backed up.

## Migrating to an Azure deployment

You can migrate your existing JIRA Data Center instance into Azure. To do this, you will need to set up a new JIRA Data Center instance in Azure, and then import data from your existing instance. This approach ensures that your new site is created with optimum settings for Azure.

For an overview of steps, see [Getting started with JIRA Data Center on Microsoft Azure](#).

## Upgrading

Before you begin, consider upgrading to an [Atlassian Long Term Support releases](#) (if you're not on one already). Long Term Support releases get fixes for critical bugs and security issues throughout its two-year support window. This gives you the option to keep a slower upgrade cadence without sacrificing security or stability. Long Term Support releases are suitable for companies who can't keep up with the frequency at which we ship feature releases.

Here's some useful advice for upgrading your deployment:

1. Before upgrading to a later version of JiraData Center, [check if your apps are compatible](#) with that version. [Update your apps](#) if needed. For more information about managing apps, see [Using the Universal Plugin Manager](#).
2. If you need to keep Jira Data Center available to users during your upgrade, we recommend [upgrading Jira Data Center with zero downtime](#). This method allows your nodes to work on different Jira versions while you upgrade them one by one. During the upgrade, Jira remains available to users.
3. We strongly recommend that you perform the upgrade first in a *staging* environment before upgrading your production instance. [Creating a test environment for Jira](#) provides helpful tips on doing so.

## Upgrading the node's operating system

The easiest way to upgrade the node's operating system is to reimage the node. It will be terminated, and will come back running the latest OS.

1. In Azure Portal, access the **JIRAcluster** Virtual Machine Scale Set (VMSS).
2. Click **Instances**.
3. Select a node, and click **Reimage**.

## Upgrading Jira Data Center on Azure

You can upgrade your whole Jira Data Center cluster to the latest version. The steps described below are similar to [Upgrading JIRA Data Center with zero downtime](#), so you should use this as a reference for any details.

The upgrade process will include putting your JiraData Center into upgrade mode so your nodes can work on different Jiraversions. Then, you will terminate your nodes and start the new ones using the latest available Jiraversion. Your cluster will remain active during the upgrade, which means that your users can still use Jirawhile you upgrade.

1. Put Jirainto upgrade mode to allow your nodes to work on different JIRA versions. Go to **Applications > JIRA upgrades**, and click **Put JIRA into upgrade mode**.
2. Scale your cluster down to 1 instance.
  - a. In Azure Portal, access the **JIRAcluster** Virtual Machine Scale Set (VMSS).
  - b. Go to **Scaling**, and scale down to 1 instance. This will delete all of your remaining nodes.
3. Get your Azure deployment to use the latest Jiraversion.
  - a. In Azure Portal, access the shared home directory. You can find it in your resource group, called something like **JIRAstorage<id>**.
  - b. Click **Files**, and then open **JIRA-shared-home**.
  - c. Delete the **JIRA-software.version** file. This file controls the Jiraversion your cluster is using.
4. Scale your cluster up by 1 instance.
  - a. In Azure Portal, access the **JIRAcluster** Virtual Machine Scale Set (VMSS).
  - b. Go to **Scaling**, and scale up to 2 instances. This will spin up a new node, but this time it will download the latest available Jiraversion.
  - c. *(info)* At this point, your cluster should be running in the mixed node, which means that your nodes are running different Jiraversions. You can check that in Jiraby going to **Applications > JIRA upgrades**.
5. Reimage the original node, so it can also download the latest Jiraversion.
  - a. In Azure Portal, access the **JIRAcluster** Virtual Machine Scale Set (VMSS).
  - b. Go to **Instances**.
  - c. Select the node, and click **Reimage**.
  - d. Make sure the node started successfully and joined the App Gateway's healthy backend pool.
6. Finalize the upgrade.
  - a. In Jira, go to **Applications > JIRA upgrades**. You should now be able to run upgrade tasks.
  - b. Click **Run upgrade tasks**. Your cluster should now be back to the normal mode.
7. If you're using more nodes than just 2, increase their number in the Virtual Machine Scale Set (VMSS).

# Federating Jira - managing multiple instances

If your organization uses multiple instances of Jira to handle the load of your users, we recommend that you consolidate those instances into a single [Jira Data Center instance](#). This will provide you with the administrative features, interoperability, high availability, and performance every large organization needs from Jira.

However, we recognize that consolidation and migration might not be a viable option for some organizations. If you're one of them, then you may need to keep your instances *federated* this means keeping them autonomous, yet connected and interoperable. This guide explains the range of features and practices to help keep your federated instances healthy.

 Some of the apps (formerly known as add-ons) mentioned in this guide are not developed by [Top Vendors](#), and some of the practices may not suit your organization's setup.

## What leads to a federated environment

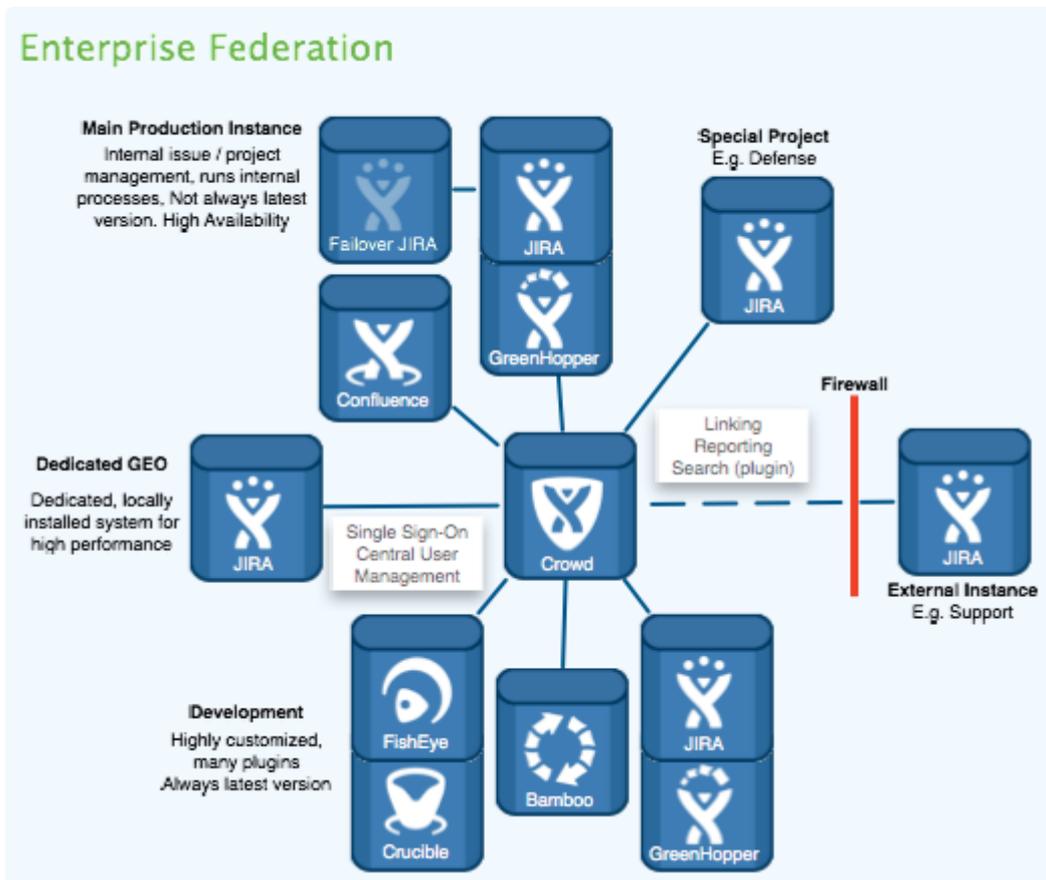
First of all, let's look at the circumstances that lead to a federated environment. When talking to our customers, we can identify four main reasons you may wish to run multiple instances of Jira:

- **Bottom-up growth of Jira:** Due to Jira's low price point and practical value, Jira often starts in a single team and then spreads throughout an organization with new teams spinning up their own server instances.
- **Mergers & acquisitions:** Through acquiring or merging with other organizations, a company can find itself managing several Jira servers.
- **Autonomous IT organizations:** Different departments within an organization may run their own IT organizations. This leads to parallel Jira systems with possible integration at a later stage once cross-divisional processes and collaboration is encouraged.
- **Intentional federation set up:** One, or many, reasons listed below could lead to an intentional set up of federated Jira systems.

## Reasons to set up a federated environment

Delving into the intentional set up of dedicated instances, we can identify several reasons for doing so by summarizing the practices of our customers, partners and our own IT organization:

- **Public / Private Jira:** Jira offers the functionality to make specific content only accessible to identified groups through permission schemes and user groups. However, many customers, including Atlassian itself, prefer to run internal content (e.g. development) on a separate instance inside the firewall. A dedicated instance is then set up to host external content (e.g. support / public feature requests) and make it accessible to customers and partners.
- **Different grades of complexity:** Let's look at an example where a Jira instance has a single project that targets a minority of users, yet is responsible for the majority of complexity, such as custom fields, workflows and customizations. To prevent this complexity from affecting the overall performance and stability of the entire Jira instance, it makes sense to migrate that project to a dedicated Jira server.
- **Access to new features vs stability:** An organization might have a situation where a specific teams' productivity is heavily influenced by their access to new features (e.g. new Jira Software features for Agile teams), while other parts of the organization have implemented stable, specific workflows and value system stability over new features. In this case, it might be a viable option to run a stable, conservative, production instance while operating a separate productive Jira server for early adopters.



- **External Customer requirement:** Some customers might require an organization to run their projects and host their data on systems that are physically separated from the rest of their infrastructure. Such requirements can be the case with classified projects, such as defense contracting.
- **Different Configuration or Settings:** Jira is famous for its configuration capabilities offering schemes for almost every concern of the application. This enables one instance to host a myriad of different projects. This is useful in cases where a project requires a dedicated instance so that it can have its own branding, or different settings. These might include separate priorities or resolutions, a different set of administrators, use of time tracking or not, and everything else that is a global setting in Jira.
- **Legal requirements:** Legal concerns, such as data privacy laws, jurisdiction over servers or government surveillance might lead to the decision to have dedicated servers in separate jurisdictions for specific Jira projects. An example of such a case would be the inability of a European company to store private data of European Union citizens on servers that are hosted in countries outside the EU that do not offer data privacy laws with the same protection of individual rights.
- **Jira as an internal service:** Some of our large customers offer Jira internally as a service. They have a standard configuration that they then clone for each business unit that requests a Jira system. Each system then has its own BU-specific administrators while the central IT organization maintains system administration competency.
- **Spreading Risk / Downtime:** Larger instances take longer to upgrade. Therefore, one possibility to reduce unavailability of Jira services to all users is to split and thereby reduce the size of instances. That way, an upgrade downtime of eight hours could then be split and turned into two-hour downtimes of four Jira systems with each downtime affecting less users.

As an example of federation, let's share with you how Atlassian deploys dedicated instances for various purposes:

Name	Purpose	Main reason to run on dedicated instance	Example Application links

JAC	Public facing issue management instance that includes new feature requirements, change requests and bugs. The best feature here is the possibility for customers to vote on issues.	Public / Private	Linking between JAC issues to related development issues in JDOG.
SAC	External support instance. Issues are visible to the reporter (support requester) and Atlassian only due to the classified nature of included information.	External Customer requirement , Public /Private	Linking between SAC support cases with related JAC issues (visible to the support customer) and JDOG (visible to Atlassian only).
JDOG	Jira teams' dogfooding server - validates your notion of having something on the cutting edge of features (for us, this is a very unique case since Jira is our product, but you get the idea).	Access to new features	Linking back to JAC and SAC respectively.
HelloJ	Our internal Jira that runs our business process, from marketing to procurement and internal IT.	Public / Private	

## Recommendations for managing multiple Jira instances

Federation of Jira instances is quite common among enterprise customers. Past research by Atlassian counted hundreds of customers that run three or more Jira servers concurrently within their organization. As a result, customers with federated environments are not alone, but part of a larger community. Atlassian has therefore developed and is continuing to implement features that make management and interoperability of federated Jira instances easier for customers.

### Federation features

The ability to have Atlassian products work together is one of the strongest features of Jira, Confluence, and the entire Atlassian stack. One of the best examples of this is the possibility to dynamically list Jira issues within a Confluence page. This same functionality also enables the integration of autonomous Jira instances. The Jira instances, thereby, do not need to run the exact same version as long as all versions contain the Application Links functionality. This makes interoperability in a heterogeneous landscape (e.g. see reasons to federate, "[Access to new features](#)") a lot easier.

In a nutshell, the goal of application links is to make inter-instance linking and integration of issues as easy as inter-project integration within the same instance! So after an application link between your Jira instances is [set up](#), the integration functionality described below becomes available:

#### Unified activity streams

Activity streams not only show updates from the instance you are on but also from connected Jira or other Atlassian product instances (e.g. Confluence or Fisheye). For example, two companies working together on a project and linking up their Jira instances could be updated with new status changes of work at their partner's Jira through a unified activity stream.

#### Unified reporting dashboards

[Gadgets](#) that draw their data from another linked and trusted Jira instance can be added to a Jira dashboard. This enables a unified reporting overview of all relevant instances. Next to the standard gadgets, you can also [add additional gadgets](#). The applications are endless from just one gadget that reports on a specific metrics of another project in another instance, to having one central instance in your federation that runs reporting of all projects in all Jira systems!

#### Remote issue links

Linking issues is crucial to connecting relevant Jira issues. With [remote links](#), you can not only track dependencies and other relations of issues to other local projects, but to remote projects in other instances as well.

### Application navigator

You do not have to remember the addresses of your Atlassian products any more. The [Application Navigator](#) header allows for fast switching between Jira instances, as well as other Atlassian products working together.

### Jira Issue Copy (by [Atlassian Labs](#), not supported)

[Atlassian Labs](#) has developed the Jira to Jira Issue Copy, which allows users to copy an issue from one Jira instance to another. It is currently still in Atlassian Labs and [available on the Atlassian Marketplace](#). It has a cleaner and easier to understand user interface, and supports more custom configurations and enhancements in the user matching algorithm. Note that this add-on is **not** supported by Atlassian, and although it may work for you now, we don't recommend building it into a process that you rely on due to its [Atlassian Labs](#) status.

### Federated search

Although searching across multiple instances is not part of an out-of-the-box Jira installation, you can achieve it through [ALM's Jira Client](#). This app is a desktop application that was designed to enable offline work with Jira. It lets you establish connections to an unlimited amount of Jira instances and once this connection has been established, search queries are executed automatically across all the various systems.

## Administering federated instances

The following sections describe Jira features that you can use to efficiently administer a federated Jira instance:

### Central user management

Jira offers multiple ways to have a single-sourced user management, group definition and authentication (SSO) that multiple Jira instances or even other Atlassian products can draw from:

- **Crowd:** Atlassian's dedicated product to provide centralized identity management with single sign-on for an organization's applications and developer tools. See the chapter on [Connecting to Crowd for User Management](#) for more instructions. In case you want to migrate user directories from Jira to Crowd, see [Importing Users from Jira to Crowd](#).
- **LDAP:** Is the *de facto* standard on storing and managing distributed directory information. See the [Jira documentation](#) on how to [connect to an LDAP Directory](#).

For an overview of different scenarios on how multiple Jira systems and/or other Atlassian products could interoperate and accomplish centralized user management, see illustrated [Diagrams of Possible Configurations for User Management](#).

### Permission scheme harmonization

The ability to have a single source for user groups also facilitates the harmonization of permission schemes across multiple systems. Groups such as Project Administrators, System Administrators, Internal, External, etc. can be centralized and then assigned to Jira's local permission schemes.

### Workflow sharing

The [Workflow Sharing](#) feature allows you to share your team's workflow with other teams in your organization on different Jira instances, or external parties in other organizations via the [Atlassian Marketplace](#). This feature allows you to easily use workflows that other people have published, or to move a workflow from staging to production in your own organization. It is also provided as an [app](#) for older Jira releases.

### Migrate, merge and split

In a federated environment, multiple projects run on multiple instances of Jira. In certain circumstances, it might be desirable to change the federation landscape by altering the number of instances involved or moving projects between them. This can be done by merging and splitting instances as well as through project migration.

**⚠️ Test first!** For all the procedures below, we highly recommend performing this procedure on a [staging environment](#) first before production systems are touched.

- **Migrate a project to another instance - CSV import:** If you only want to migrate one project, a quick way is to use the CSV Importer:
  - a. Create the project in the target instance.
  - b. Export the issues in the source project to a CSV Format.
  - c. Import it into the target system using the CSV importer and use the wizard to match meta-data to the target instance configuration.

**⚠️** Please note that a CSV import will only migrate the issue data itself. App data stored in active objects will not be carried over.

For more information, please see [Importing Data from CSV](#).

- **Merge / Migrate - project import:** One way to migrate a single project is to use the export function in the source system and then use project import in the target Jira instance.
  - a. Make sure that the two instances run the same versions of the Jira and any apps.
  - b. Also ensure that the custom fields, user groups, entities like priorities and statuses as well as schemes related to the projects in the source system exist in the target system with the same nomenclature.
  - c. Perform a full or partial XML export from the source system.
  - d. Import the XML files into the target system.

See [Restoring a Project from Backup](#).

- **Splitting an instance - full XML backup restoration:** Using the restoration feature, it is possible to clone a Jira instance through a backup/restore procedure.
  - a. Set up a new Jira Data Center with the same version of the existing system. **ⓘ** This will require a Data Center license.
  - b. Perform a full backup of the source system.
  - c. Import the backup into the new Jira instance.
  - d. Choose which duplicate projects to remove from the source and target instances.
  - e. Please note that this option only applies if the target instance is **empty**.

For more detailed instructions, see [Splitting Jira applications](#).

**ⓘ** Although it is possible to [run multiple Jira instances on the same server](#), dedicated hardware / VM is recommendable for enterprise use of Jira.

- **Splitting an instance - DB / Filesystem level:** Some apps may use the Jira database in a way that can not be exported through the standard XML export.
  - Set up another Jira instance and replicate the application server, database and installation / home directories. For detailed instructions, see [staging environment](#).
  - The steps here are essentially the same as the above: *Splitting an instance - full XML backup restoration*, however this time use your native database backup tools.

### Configuration management

The more Jira instances a federated system has the more there is a need for central administration of these systems, we see clients achieving configuration management by implementing various methods:

- **Version Control System:** A VCS like Subversion or GIT can be used to keep track of changes to files such as configuration XMLs or templates and can then be deployed to a [staging environment](#) or production system respectively. This approach provides the ability to roll-back changes and having one central overview of all configuration files in all instances.
- **Configuration Management Tool:** Many of our customers use configuration management software to keep track of their IT systems, versions, etc. An example would be [Puppet](#) in combination with [puppet-Jira app](#).
- **Continuous Deployment:** CI applications like Bamboo can be used to automatically or manually deploy configuration changes to a Jira's [staging environment](#)/ production system.

- **Jira Workflow Sharing Plugin**, by *Atlassian Labs (not supported)*. Also mentioned above in the 'Merge/Split' section. Can be used to centrally maintain and configure workflows and share them across multiple Jira systems.



The [Jira Auditor](#) app maintains an audit log on configuration changes in Jira. This is especially important when multiple administrators make changes to the same instance. Alternatively, Jira Data Center 8.8 + offers Advanced audit log to track all the events on your Jira instance.

### Agile boards in distributed environment

If you need to work in distributed environment [WatchTower for Jira](#) can help you to consolidate issues from remote Cloud and Server Jira instances into one single agile board. WatchTower gives you a single point of control to work with issues from remote instances on agile board in your Jira. Watchtower saves you time on context switching and gathering complete picture for several projects spread over federated environment.

### Synchronization of projects across instances

In some scenarios, when Jira systems are integrated and deal with similar entities, it might be desirable to have actions in one system affect another. This is not an out-of-the-box feature but can be achieved through the use of apps and scripting:

- **Backbone Issue Sync for Jira**, supported by *K15t Software GmbH*  
The [Backbone Issue Sync for Jira](#) synchronizes issue data between different Jira instances. Synchronize Jira project data across departmental and B2B borders with ease, flexibility, and confidence.
- **JJUPIN**- Simple Issue Language (SIL) for Jira, supported by *Kepler-Rominfo*  
The [JJUPIN app](#) provides a new scripting layer for Jira that enables administrators to create new functionality. It is discussed in the marketplace as being a solution for synchronization and cross-system workflows for federated Jira environments.
- **Jira Enterprise Mail Handler**, supported by *Javahollic Software*  
The [Jira Enterprise Mail Handler](#) is an app which was originally designed to provide advanced functionality to handle incoming emails. This includes the interaction with users that do not have a Jira account or users that use several email aliases. Through its project and field mapping, it can also be used to synchronize and update issues between different instances. For example, you can automatically create an issue in instance A if the issue transitions to a particular state in instance B. Setting up such automation however should be thoroughly tested as there can be occurrences of infinite loop if issues are updated at the same time on both instances.
- **ScriptRunner**, supported by *Adaptavist*  
[ScriptRunner](#) is used by many customers to add further scripting functionality to Jira. The app supports input in languages including Groovy, Python (jython), Ruby (JRuby) and JavaScript.

### Upgrading multiple instances

In order to avoid manual repetition of the [upgrade instructions](#) given in the Jira administration manual, there are ways to partially or fully automate the process:

- **Automated Backup**: You can export data from Jira using the [Jira Command line interface](#). (The Jira CLI can also be quite useful for automated tests).
- **Symlinks**: Instead of working only with the installation directory, a system administrator could maintain a directory for each Jira version. Once the new version has been installed and modified, a symbolic link could be modified to point to the folder of the new version. This is a practice that has proven itself at Atlassian internally.

For Jira instances with high availability requirements, it is also recommended to use a [staging server setup for Jira](#). You do not require a separate license for a staging server.

As discussed in the [Application Links](#) section, Jira instances don't have to run the same version in order to enable application links.

### Planning HA / failover for multiple instances

Federating Jira instances on one Jira Data Center will enable you to set different availability requirements for your instances. You can certainly leverage the same tools for many instances such as the load balancer (given it runs in HA mode as well) for multiple Jira instances.

Please refer to the [Failover for Jira Guide](#) for more information.

## More resources

We have a range of services and programs designed to help you choose and implement the right solution for your organization. Check our [Enterprise services](#) page for details about services included with your Data Center license, as well as our paid services.

# Layout and design

The layout and design of your Jira applications can be customized to an extent, so that they more closely reflect your organization's look and feel, and requirements in relation to user preferences (such as the default language, user preferences and announcement banner).

Search the topics in 'Layout and design':

The following pages provide you with more information on setting up your Jira applications so you can get the most out of them!

- [Configuring the look and feel of your Jira applications](#)
- [Configuring an announcement banner](#)
- [Configuring the default dashboard](#)
- [Choosing a default language](#)
- [Configuring the default issue navigator](#)
- [Creating links in the application navigator](#)
- [Configuring the user default settings](#)



You may also wish to extend Jira's functionality by installing and/or enabling new plugins. Read the [Managing apps](#) page for further information.

# Configuring the look and feel of your Jira applications

As a Jira administrator, you can customize the look and feel of your Jira applications to match your company's environment. This page will walk you through the following:

- [How to change your logo](#)
- [How to show your site title](#)
- [How to change the favicon](#)
- [How to change Jira application colors](#)
- [How to change the gadget colors](#)
- [How to change date and time formats](#)

## How to change your logo

The logo appears in the top left corner of every Jira application page. The height of the logo image must be constrained to 30 pixels. We also recommend you use an image with a width of 57 pixels.

1. In the upper-right corner of the screen, select **Administration**  > **System**.
2. In the User interface section, click **Look and feel**.
3. In the Logo section, upload the image for the logo you want to use in your Jira application. You can also upload from a URL beginning with 'http://' or 'https://'.

## How to show your site title

If enabled, your site name will appear next to your logo.

1. In the upper-right corner of the screen, select **Administration**  > **System**.
2. In the User interface section, click **Look and feel**.
3. Select the **Show Site Title** checkbox to make your instance name appear next to your logo in the header.

## How to change the favicon

The favicon appears typically to the left of your browser's URL field, and on browser tabs displaying a page on your Jira site. To upload a favicon, make sure it's in PNG format, with dimensions of 32x32 pixels, 71x71 DPI, and with 8-bit color depth.

1. In the upper-right corner of the screen, select **Administration**  > **System**.
2. In the User interface section, click **Look and feel**.
3. In the Favicon section, upload the image for the favicon in your Jira application or upload from a URL beginning with 'http://' or 'https://'.

## How to change Jira application colors

You can use color options to control the appearance of the entire Jira user interface. The colors you choose for each option can be anything that is valid for both a font tag, and a stylesheet's 'color' attribute.

1. In the upper-right corner of the screen, select **Administration**  > **System**.
2. In the User interface section, click **Look and feel**.
3. In the Colors section, modify the color schemes for the different elements of your Jira instance as needed using the pop-up color chooser, or by specifying your own (eg. #FFFFFF, 'red').
4. To return to the original color scheme, just clear any values that you've set, or click **Revert** in the row where you made the change.

## How to change the gadget colors

You can use any of the color options to change the color of a gadget's frame on the Jira dashboard.

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. In the User interface section, click **Look and feel**.
3. In the Gadget Colors section, select the color option for the gadget's frame on your Jira dashboard, and then modify the color as needed using the pop-up color chooser, or by specifying your own (eg. '#FFFFFF', 'red').
4. To return to the original color scheme, just clear any values that you've set, or click **Revert** in the row where you made the change.

 Color 1 is the default frame color for newly-added gadgets. The colors you specify for each of the options can be anything that is valid for both a font tag, and a stylesheet's 'color' attribute.

## How to change date and time formats

You customize the way times and dates are presented to users throughout the Jira user interface. When specifying dates and times, they should be based on the [Java SimpleDateFormat](#).

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. In the User interface section, click **Look and feel**.
3. In the Date/Time Formats section, click the value of the element you want to configure, then update the value as necessary.

 Issue date/time fields show a relative instead of absolute date/time format. For example, "Yesterday" would appear instead of "20 May 2013 12:00 PM". You can still see the absolute date/time by hovering over the field. The date/time format reverts to absolute after a week.

Here are some further examples of US date/time configurations:

Preferred Date/Time	Value of the <code>jira.date.time.picker.java.format</code> property	Value of the <code>jira.date.time.picker.javascript.format</code> property
2010-10-15 08:50	yyyy-MM-dd HH:mm	%Y-%m-%d %H:%M (see <a href="#">ISO 8601 format</a> )
15/Oct/10 8: 50 AM	dd/MMM/yy h:mm a	%d/%b/%y %l:%M %p
10/15/10 08: 50 AM	MM/dd/yy hh:mm a	%m/%d/%y %I:%M %p

# Configuring an announcement banner

System Administrators can configure an announcement banner to display pertinent information on all Jira pages. The banner can be used to relate important information (e.g. scheduled server maintenance, approaching project deadlines, etc.) to all users. Further, the banner visibility level can be configured to display to all users or just logged-in users.

If you're using Jira Server, the banner can be configured to contain HTML text.

## On this page:

- [Configuring an announcement banner](#)
- [Banner visibility mode](#)

 For all of the following procedures, you must be logged in as a user with the **Jira System Administrators** [global permission](#).

## Configuring an announcement banner

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. Select **User interface** **> Announcement banner** in the System panel below.
3. Enter the required text in the **Announcement** field.
4. Select the required **Visibility level** for the banner.
5. Select the **Set banner** button.

Depending on the visibility level selected, the banner will become visible throughout Jira.

## Banner visibility mode

The announcement banner visibility level can be configured to specify to whom the banner will be displayed. There are two modes:

- **Public** the banner is visible to everyone
- **Private** the banner is visible to logged-in users **only**.

# Configuring the default dashboard

The default dashboard is the screen that all Jira users see the first time they log in. Any users who have not added any dashboard pages as favorites also see the default dashboard.

Jira allows Administrators to configure the default dashboard. The gadgets on the default dashboard can be re-ordered, switched between the left and right columns, additional gadgets can be added, and some gadgets can be configured. The layout of the dashboard (e.g. number of columns) can also be configured.

All changes made to the default dashboard will also change the dashboards of all users currently using the default. However, gadgets that users do not have permissions to see will not be displayed to them. For example, the 'Administration' gadget, although it may exist in the default dashboard configuration, will not be visible to non-admin users. [Gadgets](#) are the information boxes on the dashboard. Jira comes pre-configured with a set of standard dashboard gadgets. It is also possible to develop custom gadgets and plug them into Jira using its flexible [plugin system](#).

**Note:** For all of the following procedures, you must be logged in as a user with the [Jira Administrators global permission](#).

## Adding and configuring gadgets on the default dashboard

Jira's default dashboard is limited to only one dashboard page. However, users can add multiple pages to their own dashboards if they wish.

1. In the upper-right corner of the screen, select **Administration**  **System**.
2. Select **User Interface > System dashboard** to open the Configure System Dashboard page.
3. On the 'Configure System Dashboard' page, you can do the following:
  - Move a gadget by drag-and-drop.
  - Re-configure existing gadgets.
  - Choose a different layout.

**i** By default, there is a limit of 20 gadgets per dashboard page. If you wish to raise this limit, edit the [jira-config.properties](#) file, set `jira.dashboard.max.gadgets` to your preferred value and then restart Jira.

## See also

- [Using dashboard gadgets](#)
- [Adding a gadget to the directory](#)
- [Subscribing to another application's gadgets](#)

# Using dashboard gadgets

On this page:

- [About gadgets](#)
- [Pre-installed gadgets](#)
- [Extension gadgets](#)
- [Creating new gadgets](#)

## About gadgets

Gadgets display summaries of Jira project and issue data on the [dashboard](#). You can customize gadgets to display project and issue details relevant to particular users.

### Adding Atlassian gadgets to external websites

You can also add Atlassian gadgets to compatible external websites, like iGoogle. For instructions on how to do this, refer to [Adding an Atlassian Gadget to iGoogle and Other Web Sites](#).

## Pre-installed gadgets

Jira provides a set of standard gadgets out-of-the-box:

Gadget	Description
Activity Stream Gadget	Displays a summary of your recent activity.
Administration Gadget	Displays checklist of common administration tasks and links to administrative functions and documentation.
Assigned To Me Gadget	Displays all open issues in all projects assigned to the user who views the dashboard.
Average Age Gadget	Displays a bar chart of the average number of days that issues have been unresolved.
Bamboo Charts Gadget*	Displays charts and plan statistics from a Bamboo server.
Bamboo Plan Summary Chart Gadget*	Displays a graphical summary of a build plan.
Bamboo Plans Gadget*	Displays a list of all plans on a Bamboo server, and each plan's current status.
Bugzilla ID Search Gadget	Allows the user to search all Jira issues for references to Bugzilla IDs.
Calendar Gadget*	Shows issues and versions in a calendar format based on their due date. Calendars can be based on an issue filter or on a project.
Clover Coverage Gadget*	Displays the Clover coverage of plans from a particular Bamboo server.

Created vs Resolved Gadget	Displays a difference chart of the issues created vs resolved over a given period.
Crucible Charts Gadget*	Displays various charts showing statistical summaries of code reviews.
Favorite Filters Gadget	<p>Displays a list of the favorite filters for the user viewing the dashboard.</p> <div style="border: 1px solid red; padding: 5px; margin-top: 10px;">  If your Jira instance has many users and favorite filters, this gadget might cause JIRA to be slower than usual. </div>
Filter Results Gadget	Displays the results of an issue filter.
FishEye Charts Gadget*	Displays two charts showing showing statistics about a sourcecode repository.
FishEye Recent Changesets Gadget*	Displays a number of recent changesets from a FishEye repository.
In Progress Gadget	Displays all issues that are in progress and assigned to the user viewing the dashboard.
Introduction Gadget	Displays a configurable introduction message on the dashboard.
Issue Statistics Gadget	Displays the collection of issues returned from a filter, broken down by a field.
Pie Chart Gadget	Displays issues from a project or issue filter, grouped by a statistic type, in pie-chart format. Issues can be grouped by any statistic type (e.g. Status, Priority, Assignee, etc).
Projects Gadget	Display information and filters related to a project(s).
Quick Links Gadget	Displays useful links to issues associated with the current user.
Recently Created Issues Gadget	Displays a bar chart of the rate at which issues are created, as well as how many of those issues are resolved.
Resolution Time Gadget	Displays a bar chart of the average resolution time (in days) of resolved issues.
Road Map Gadget	Shows versions which are due for release within a specified period of time, and a summary of progress made towards completing the issues in those versions.
Text Gadget*	Displays configurable HTML text on the dashboard.
Time Since Issues Gadget	Displays a bar chart of the number of issues that something has happened to within a given time period. The 'something has happened' is based on a date field that you choose, such as 'Created', 'Updated', 'Due', 'Resolved' or a custom field.

Two Dimensional Filter Statistics Gadget	Displays tabular data based on a filter.
Voted Gadget	Shows issues for which you have voted.
Watched Gadget	Shows issues you are watching.

📘 For more ideas, see the [big list of all Atlassian gadgets](#).

\*This gadget is only available if you have installed/configured the relevant app.

## Extension gadgets

Other gadgets are available as apps on the [Atlassian Marketplace](#). To use these apps, install and [enable](#) them.

## Creating new gadgets

You can create new gadgets by writing an XML descriptor file, packaged as an [Atlassian app](#). See [Developing Gadgets](#) for more information.

Related topics

[The big list of Atlassian gadgets](#)

# Adding a gadget to the directory

The Jira gadget directory displays all the gadgets that are available for Jira users to add to their dashboard.

 For all of the following procedures, you must be logged in as a user with the **Jira System Administrators** [global permission](#).

## On this page:

- [Adding a Gadget that is Not a Plugin](#)
- [Adding a Gadget that must be Installed as a Plugin](#)

## Security implications

Add only gadgets from sources that you trust. Gadgets can allow unwanted or malicious code onto your web page and into your application. A gadget specification is just a URL. The functionality it provides can change at any time.

There are two types of gadgets: those that must be installed as plugins, and those that can be added as simple gadget URLs.

## Adding a Gadget that is Not a Plugin

If the gadget is hosted on another server and can be added to the directory as a simple URL, then you can simply add it via your dashboard's 'Add Gadget' option.

**To add a gadget to your directory,**

1. First you need to find the URL for the gadget's XML specification file. Gadget authors and publishers make their gadget URLs available in different ways. Below are the instructions for an Atlassian gadget and a Google gadget.

- Follow the steps below if you need to find the URL for a gadget that is published by an Atlassian application, such as JIRA or Confluence: A gadget's URL points to the gadget's XML specification file. Gadget URLs are shown on the '**Gadget Directory**' screen that is displayed when you click '**Add Gadget**'. In general, a gadget's URL looks something like this:

```
http://example.com/my-gadget-location/my-gadget.xml
```

If the gadget is supplied by a plugin, the URL will have this format:

```
http://my-app.my-server.com:port/rest/gadgets/1.0/g/my-plugin.key:my-gadget/my-path/my-gadget.xml
```

For example:

```
http://mycompany.com/jira/rest/gadgets/1.0/g/com.atlassian.streams.streams-jira-plugin:activitystream-gadget/gadgets/activitystream-gadget.xml
```

To find a gadget's URL in JIRA:

- Go to your dashboard by clicking the '**Dashboards**' link at the top left of the screen.
- Click '**Add Gadget**' to see the list of gadgets in the directory.
- Find the gadget you want, using one or more of the following tools:
  - Use the scroll bar on the right to move up and down the list of gadgets.
  - Select a category in the left-hand panel to display only gadgets in that category.
  - Start typing a key word for your gadget in the '**Search**' textbox. The list of gadgets will change as you type, showing only gadgets that match your search term.
- Right-click the '**Gadget URL**' link for that gadget and copy the gadget's URL into your clipboard.

To find a gadget's URL in Confluence:

- Open the '**Browse**' menu and click '**Confluence Gadgets**' to see the list of available Confluence gadgets.
  - Find the gadget you want.
  - Right-click the '**Gadget URL**' link for that gadget and copy the gadget's URL into your clipboard.
- Follow the steps below if you need to find the URL for a Google gadget:
    - a. Go to the [Google gadget directory](#). (You can also get there by clicking 'Add Stuff' from your iGoogle home page.)
    - b. Search for the gadget you want.
    - c. Click the link on the gadget to open its home page.
    - d. Find the '**View source**' link near the bottom right of the page. Right-click the link and copy its location to your clipboard. This is the gadget's URL.
2. Now you can add the gadget to your directory. Go to the dashboard by clicking the '**Dashboard**' link or the '**Home**' link at the top left of the screen.
  3. The dashboard will appear. Click '**Add Gadget**'.
  4. The '**Add Gadget**' screen appears, showing the list of gadgets in your directory. Click '**Add Gadget to Directory**'.
-  You will only see this button if you have administrator permissions for your dashboard.
5. The '**Add Gadget to Directory**' screen appears. Type or paste the gadget URL into the text box.
  6. Click '**Add Gadget**'.
  7. The gadget appears in your gadget directory. (It will be highlighted for a short time, so that you can see it easily.)

## Adding a Gadget that must be Installed as a Plugin

If the gadget must be installed as a plugin, you cannot add it via the gadget directory user interface. Instead, you will need to follow the instructions for adding a plugin, as described in [Managing apps](#). Once you have installed your plugin, the gadget will automatically appear in the directory.

Related topics

[The big list of Atlassian gadgets](#)

# Subscribing to another application's gadgets

## Security Implications

Add only gadgets from sources that you trust. Gadgets can allow unwanted or malicious code onto your web page and into your application. A gadget specification is just a URL. The functionality it provides can change at any time.

If you have administrator privileges, you can configure your application to subscribe to gadgets from other Atlassian applications. This feature allows administrators to make all the gadgets from one application available in another application, without having to enable each gadget individually via the gadget URL.

To make use of this feature, you will need two or more applications that support the feature.

The gadgets included are those provided by the other application or via plugins installed into that application. They do *not* include external gadgets that the other application has added to its directory.

## To subscribe to gadgets from another application,

1. Go to the dashboard by clicking the '**Dashboard**' link or the '**Home**' link at the top left of the screen.
2. The dashboard appears. Click '**Add Gadget**'.
3. The '**Add Gadget**' screen appears, showing the list of gadgets in your directory. See the [gadget directory screenshot](#) below. Click '**Gadget Subscriptions**'.
  -  You will only see this button if you have administrator permissions for your dashboard, and if your application supports gadget subscriptions.
4. The '**Gadget Subscriptions**' screen appears, showing the applications to which your application already subscribes. Click '**Add Subscription**'.
5. The '**Add Subscription**' screen appears. See the [screenshot](#) below. Enter the base URL of the application you want to subscribe to. For example, <http://example.com/jira> or <http://example.com/confluence>.
6. Click '**Finished**' to add the subscription.

*Screenshot: Gadget directory with 'Gadget Subscriptions' button*

## Gadget Directory

Search

**All (67)**

[Bamboo \(8\)](#)

[Charts \(16\)](#)

[Clover \(2\)](#)

[JIRA \(53\)](#)

[Other \(4\)](#)

[Wallboard \(23\)](#)



### Activity Stream

By Atlassian

Lists recent activity in a single project, or in all projects.

<https://sgriffin.jira-dev.com/rest/gadgets/1.0/g/com.atlassian.streams.streams-jira-...>

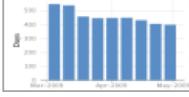


### Assigned to Me

By Atlassian

Displays all unresolved issues assigned to me

<https://sgriffin.jira-dev.com/rest/gadgets/1.0/g/com.atlassian.jira.gadgets:assigned-...>

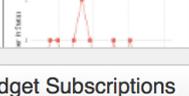


### Average Age Chart

By Atlassian

Displays the average number of days issues have been unresolved.

<https://sgriffin.jira-dev.com/rest/gadgets/1.0/g/com.atlassian.jira.gadgets:average-...>



### Average Number of Times in Status

By Atlassian

### Screenshot: Adding a gadget subscription

## Add Subscription

Type or paste the base URL of the Atlassian application to subscribe to:

Please note that subscribing to this URL will add it to your instance's whitelist. For more information on how to edit this whitelist please see the [documentation](#).

When you subscribe to gadgets from another application, people will be able to use all those gadgets on their dashboards or pages in your application. The gadgets are those provided by the other application or via plugins installed into that application. They do not include external gadgets that the other application has added to its directory.

**Only subscribe to applications that you trust! Gadgets can allow unwanted or malicious code onto your web page.**

You can subscribe to any Atlassian application that publishes its gadgets for other applications to find.

An application's base URL looks something like this: <http://example.com/jira>

Related topics

[The big list of Atlassian gadgets](#)

# Choosing a default language

## Overview

Most user-visible pages in Jira are now internationalized. When Jira is first installed, you can select from the default languages:

- Chinese,
- Czech,
- Danish,
- Dutch,
- English (UK or US),
- Finnish,
- French,
- German,
- Hungarian
- Italian,
- Japanese,
- Korean (South Korean),
- Norwegian,
- Polish,
- Portuguese (Brazilian),
- Russian,
- Spanish (Spain), and
- Swedish.

The following languages have been discontinued in Jira 8.12. You can still choose them in Jira, but new messages will be in English:

- Estonian
- Icelandic
- Romanian
- Slovak

**Note:** For all of the following procedures, you must be logged in as a user with the **Jira Administrators** [global permission](#).

## Changing the default language

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. Select **General Configuration** to open the Administration page.
3. Click the **'Edit Settings'** button, then select the appropriate language in the drop-down box next to 'Default language'.  
Any additional languages you have installed will appear in the list. See [Translating Jira](#).

## Per-user language selection

Individual users can manage their user profile, which will override the default language (see above).

## Overriding the default translations of issue types, resolutions, statuses, and priorities

Should you wish, you can easily [specify your own translations](#) for the values of the following Jira issue fields:

- Issue type
- Priority
- Status
- Resolution

Your specified translations will override the values specified in the Jira translation.

## Known issues

### On this page:

- [Overview](#)
- [Changing the default language](#)
- [Per-user language selection](#)
- [Overriding the default translations of issue types, resolutions, statuses, and priorities](#)
- [Known issues](#)
- [Related topics](#)

- When you create a project using a system language other than English, Jira will create duplicates of default issue types, statuses, resolutions, and priorities. These duplicates won't be translated into other languages chosen by your users. To work around this, either create new projects with the language set to English (and then change back to your preferred language) or provide custom translations for these duplicates (see [Translating Jira constants](#)).

#### Related topics

- [Translating Jira](#)

# Translating Jira

We're creating Jira in English, but we know it's being used by teams around the globe. Here you can learn more about the translations we provide, and find some ways to translate Jira on your own.

What translations of Jira are currently available?

## Official languages

Jira ships with a number of translations in the most commonly-requested languages, which are updated and corrected with each release. You can easily update these via the Universal Plugin Manager - see [Managing apps](#).

As a Jira administrator, you can choose the default language from the list of installed languages (see [Choosing a default language](#)). If you're a Jira user, you can also choose your preferred language in the user profile.

## Custom languages

If your preferred language is not on the list of official languages, there are a few things you can do:

- Go to [Packages Atlassian](#), and check if your language is listed there. You can download and add it to your Jira instance, or even translate some strings on your own. [This kb article](#) will help you achieve that.
- Use the [InProduct Translations](#) plugin to translate Jira on the fly, in any way you want.

## Making changes to translations

We're constantly working on improving and updating our translations. If you had a problem or just think something could be translated better, give us a shout by creating a suggestion on our public Jira instance. We'll get our best people to review and correct it.

To report a problem with translations in Jira:

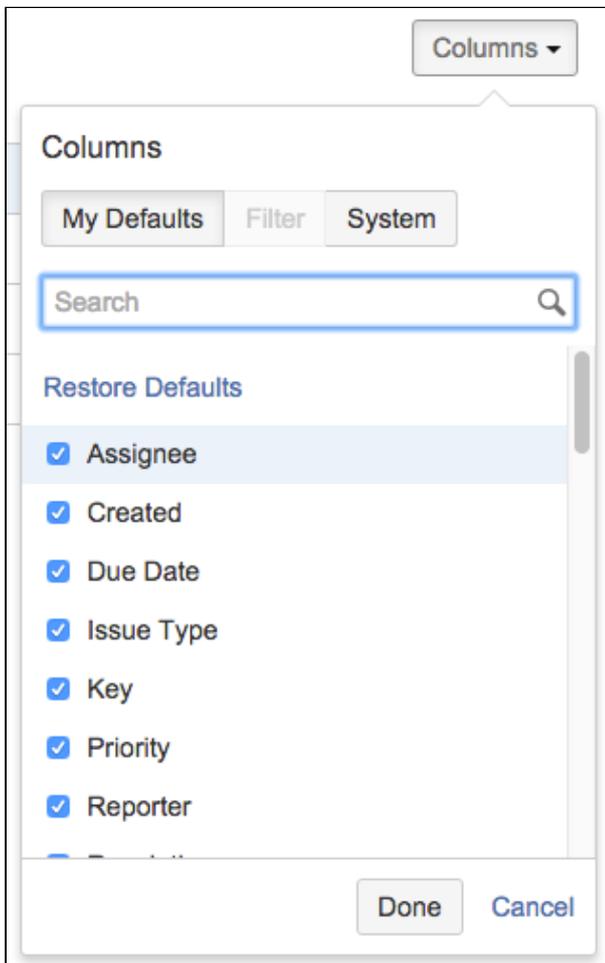
1. Go to [Atlassian Translations](#).
2. Click **Create** to create a suggestion. If possible, include the following details:
  - Product: Jira (Server)
  - Language
  - Mistranslated text. You can provide your own translation here if you want.
  - Brief explanation
  - (Optional) Screenshot
  - (Optional) Version

What about translations of the documentation?

Currently, we only offer the Japanese translation of the Jira documentation. See [Jira Core](#), [Jira Software](#), or [Jira Service Desk](#).

# Configuring the default issue navigator

Jira applications let you change the columns of the table of search results for any search results displayed using the List view. Click **Columns** at top right of the issue table to open the column configuration dialog, shown below.



*Column Configuration Dialog*

This displays the list of the columns used in the current table of results. Choose the columns you want with checkboxes and click **Done** to finish. Notice that the **Filter** option is greyed out, this is because the issue table results are not coming from a filter. See [Changing the column configuration for your own filters](#) for an example of using this dialog to set the displayed columns for your own filters.

## Sorting and rearranging columns

- To sort issues, just click on a column header.
- To rearrange the column layout, press and hold the mouse button to enter "column drag mode."

## My defaults, filter, and system

If the currently selected button is **My Defaults**, this indicates that the columns you are seeing are from your user account preferences. **Filter** is an available option whenever the issue search results come from a saved filter. If you are a Jira Admin, you will also see the **System** tab, where you can change the columns for all users who have not set their own defaults.

Jira administrators can configure the columns that appear in the Issue Navigator for all users that do not have personal column filters defined. When administrators are configuring default columns, their permissions are ignored, so that they can add a project-specific custom field from a project that they do not have permissions to browse. The field would never be actually shown to users that do not have permissions to see it. Jira administrators can also select which views are available in the Jira system, as views are configurable via [plugins](#).

**On this page:**

- [My defaults, filter, and system](#)
- [Changing the column configuration for your own filters](#)
- [Troubleshooting](#)

## Changing the column configuration for your own filters

If you are searching using a saved filter *and if the filter is owned by you*, use the **Filter** button to customize the columns displayed when users see results from that filter. When sharing a filter with other users, it's sometimes helpful to choose the relevant columns for those results. For example, if your filter searches for issues that are open bugs, you may decide to remove the columns for status and issue type for that filter since they will all be the same. Filters don't always have columns configured, but when they do, those columns will be shown unless the user chooses to use their defaults using the **My Defaults** button.

For any Jira filters that you own, you can change the displayed columns as follows.

1. Click on the name of a Jira filter you own.
2. Click the **Columns** button at top right of the currently displayed columns. This opens the column configuration dialog.
3. Select or deselect checked items in the list.
4. Click **Done** when you are finished.

## Troubleshooting

If you cannot find a column, please make sure that you haven't run in to any of the following restrictions:

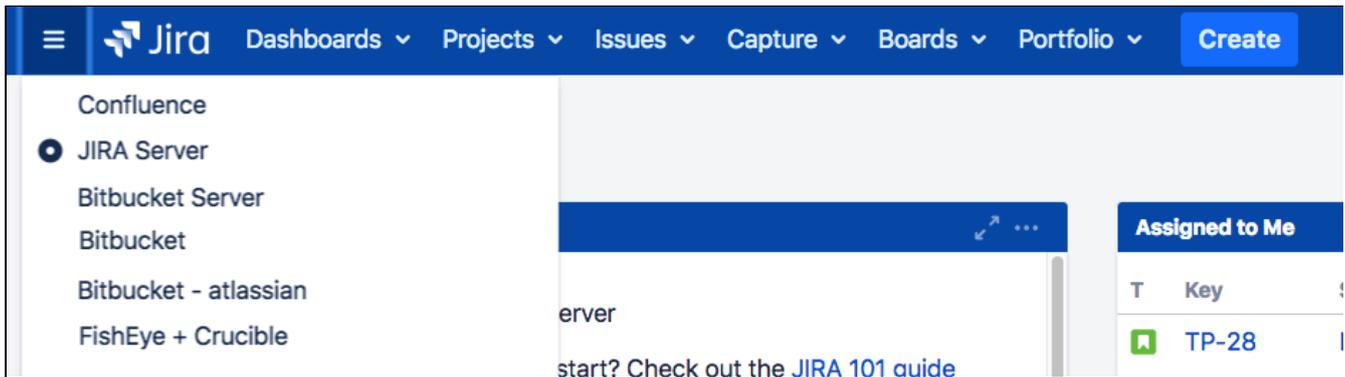
- You can only see columns for issue fields that have not been [hidden](#) and that you have [permissions](#) to see.
- It is possible to add any of the existing [custom fields](#) to the column list, as long as the fields are visible and you have the right permissions.
- Some project-specific custom fields, even if selected, do not appear in the Issue Navigator for all issues. Project-specific custom fields will be shown only if the filter has been restricted to that project only.
- Issue type custom fields aren't displayed by default in the Issue Navigator. However, if you include the issue type configured for the custom field in the query, you can select the custom field in the Column Configuration dialog to add the custom field to the Issue Navigator.

# Creating links in the application navigator

You can add custom links in the application navigator, to make it easier for users to navigate to frequently used information.

## What is the application navigator?

The application navigator is the  control in the top left of the Jira header that displays a menu of links to other applications. It is only displayed to users if there is more than one link. You can customize the links that appear in the application navigator, as well as making certain links only visible for specific users.



## Adding links to the application navigator

If applications are linked to your Jira instance via [application links](#), those applications will automatically appear in the application navigator. If you don't have any applications linked, the application navigator icon () will appear only for administrators. After links have been set up, the application navigator icon will automatically be visible to all users.

1. Select **Administration**  **Applications**.
2. Select **Application Navigator**.
3. Create links by entering a name and the URL on the page.

After you've created a link, it will appear in the application navigator for all your applications after a few minutes (up to 10). Or, if you want links to appear immediately, you can navigate to the application navigator administration page in each application and refresh the page.

If you want to make a link appear in the application navigator for only specific users, use the **Groups** box to specify which groups can see the link. To hide the link from all users, select the **Hide** checkbox (for example, if you want to temporarily hide the link without deleting it entirely).

**i** When you make a link visible for a specific group, the link visibility is only set up in the application where you are configuring the link. For example, if you change the visibility in the Jira administration screen and you also want it to be visible to the same users in Confluence, you must make the same changes in the Confluence administration settings.

To modify links that were created and are managed in other applications (for example, in a different Jira application), edit the link in that application. You cannot delete links to linked applications, you must delete the application link instead.

# Configuring the user default settings

Administrators can change the default user settings which are applied to user accounts on creation. These settings can be changed by the user on an individual basis through their profile.

**Note:** An administrator can force the user to use a specific Email format by clicking the **Apply** link. The user will then be unable to edit this setting.

## Changing the user default settings

1. Log in as a user with the **Jira Administrators** [global permission](#).
2. Select **Administration** (⚙️) > **System** > **Default user preferences** to open the User Default Settings page.
3. Click the **Edit default values** button. The User Default Settings window displays.
4. Make the changes you wish to apply. A summary of the available changes is listed below.

Setting	Option
Email format	Outgoing email notifications from Jira can be sent as HTML or text format.
Issues per page	This will set the number of issues displayed on each Issue Navigator page. Enter a value between 1 and 1000.
Default access	Choose the default access setting for when you create new filters and dashboards, which can be either shared with all other users ( <b>Public</b> ) or restricted to your viewing only ( <b>Private</b> ).
Notify users of their own changes	Choose between making Jira send you email notifications about issue updates made by either both you and other people ( <b>Notify me</b> ) or other people only (i.e. <b>Do not notify me</b> ).
Autowatch own issues	Choose between allowing Jira to automatically make you a watcher of any issues that you create or comment on.

5. Click the **Update** button. Your changes have been applied.

**Note:** The first time you access the User Default Settings window, the Email format is set to text. This will be applied if you click **Update**. Ensure you have selected the correct Email format you wish to apply.

# User management

You can use Jira to manage its own users, or you can connect Jira to an external user management system. You can also use Jira as a user management system for other Atlassian products, so that your users have the same login details for all their Atlassian products.

## Managing users

[This section](#) covers all aspects of using Jira for user management. Learn everything from how to create and view a user, to deactivation and monitoring user activity.

## Managing groups

[Learn](#) which groups exist by default when you install a Jira application as well as how to create, edit, and delete groups, add users to groups. This section also covers assigning permissions to Jira functions.

## Advanced user management

[Learn](#) about the advanced user management features in Jira, such as allowing other Atlassian products to connect to Jira for user management, enabling public signup, and user management limitations and recommendations.

## User directories

[Learn](#) more about your Jira user directory and how to connect to external directories for external user management.



Managing 500+ users across Atlassian products?  
Find out how easy, scalable and effective it can be with Crowd!  
See [centralized user management](#).

# Managing users

As a Jira administrator, you can manage users directly in Jira, or enable public signup so users can create their own accounts. You can refer to these pages for information on managing users across multiple projects and applications.

 For all of the following procedures, you must be logged in as a user with the Jira Administrators or Jira System Administrators global permission.

Documentation	What you can do
<a href="#">Create, edit, or remove a user</a>	Learn about different ways to create users in Jira, edit user information and properties, and deactivate or delete users who no longer need to use the system.
<a href="#">Assign users to groups, project roles, and applications</a>	Give users access to different functions in Jira like project roles and applications. Users are created without any access so this is a critical step to allow your team to get started.
<a href="#">Monitor a user's activity</a>	Keep an eye on user activity to keep your system running well. This information can help Administrators analyze Jira performance and also know which users have been inactive for a while.
<a href="#">Prevent automatic login</a>	Administrators can control which user login information is stored or turn off this feature completely. Read more to learn about the details and benefits.
<a href="#">Manage password policy</a>	Make sure your Jira system is secure by implementing a password policy and CAPTCHA.
<a href="#">Control anonymous user access</a>	Keep your user data safe and make sure anonymous users can't access the content they should not view.
<a href="#">Anonymize users</a>	Anonymize a user to hide or delete any data that can identify them as a real person. This can be useful if a person is leaving your organization and requests to have their personal data erased.

 Managing 500+ users across Atlassian products? Find out how easy, scalable and effective it can be with Crowd! See [centralized user management](#).

# Create, edit, or remove a user

In Jira applications, users can be managed manually or via an external user management system. This page helps you manage users manually and references external user management systems where required.

To log in and access a Jira application, a user must have application access. Users can obtain application access if they're members of a group assigned to the app. Membership in these groups can be changed at any time on a per-user basis.

## On this page:

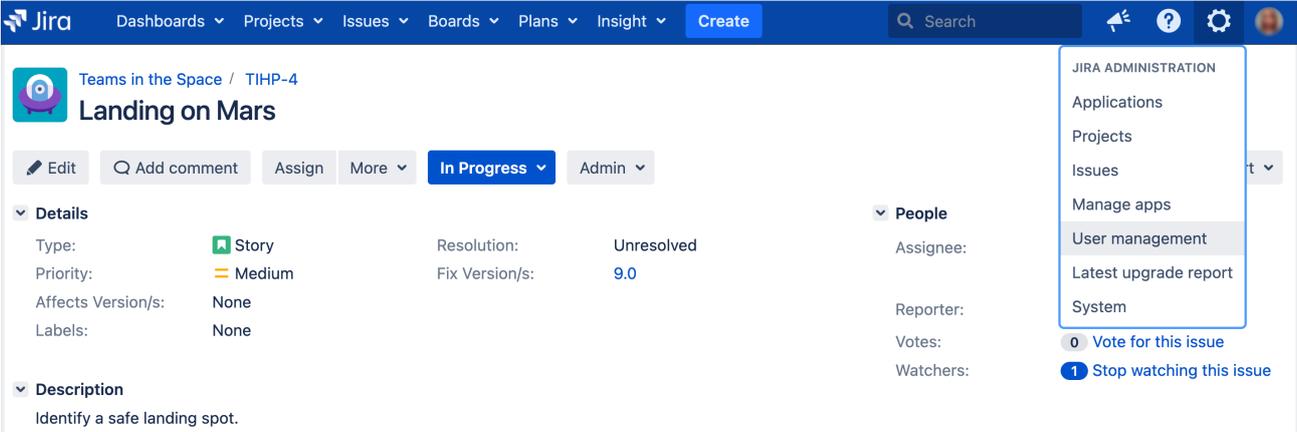
- [Before you begin](#)
- [Creating users](#)
- [Editing users](#)
- [Deactivating users](#)
- [Deleting users](#)
- [Anonymizing users](#)

**i** Managing 500+ users across Atlassian products? Find out how easy, scalable, and effective it can be with Crowd! Crowd allows you to manage users for all Atlassian Server and Data Center applications, as well as manage user authentication permissions from multiple directories. See [centralized user management](#).

## Before you begin

**i** To manage users in Jira apps, you must have the Jira Administrator or System Jira Administrator global permission. For details, see [Permissions overview](#).

To access user management settings in Jira, select **Administration**  (in the upper-right corner of the screen) > **User management**.



The screenshot shows the Jira interface for a project named 'Teams in the Space / TIHP-4' with the issue 'Landing on Mars'. The 'Administration' menu is open, showing options like Applications, Projects, Issues, Manage apps, User management (highlighted), Latest upgrade report, and System. The 'People' section shows the Assignee, Reporter, Votes, and Watchers.

## Creating users

There are several ways to create a user in Jira. Read on to learn which method is right for your team.

**i** If you're adding users to a Jira Service Management project, check out [Setting up users](#).

## Create users manually in Jira

If you have a small team, you can create users directly in Jira.

**i** Consider external user management (LDAP or Active Directory) if you have a lot of hands on deck. Maintaining permissions for individual user ID's can be messy if you have too many users, so there are other options for your large staff. See [Create users automatically](#) for more details.

1. In the upper-right corner of the screen, select **Administration**  **> User Management**.
2. In the Userbrowser that opens, select **Create User** button (located in the upper-right corner of the screen).
3. Enter the **Username**, **Password**, **Full Name**, and **Email** address.
4. Optionally, select the **Send Notification Email** checkbox to send the user an email containing:
  - their username
  - a 24-hour valid link from which they can set a password
5. If you have more than one Jira application installed, you will need to select the Jira application you want to give the user access to.
6. Select **Create another** if you want to create more users.
7. Select **Create user** to save your changes.

After the user is created, you'll be brought to the User browser where you can view the user information and perform additional functions such as edit details, edit groups or properties, and delete the user.

### Invite users to Jira via email

You can invite users to Jira through email. When they accept the invitation, they will be given access to the [applications set as default](#).

 Jira's [SMTP mail server](#) must be configured to send notifications before you can invite users via email.

1. In the upper-right corner of the screen, select **Administration**  **> User Management**.
2. In the Userbrowser that opens, select **Invite Users** button (located in the upper-right corner of the screen).
3. Enter the email addresses of the users that you want to invite. To add multiple users, separate their email addresses with a comma.
 

**Note:** You cannot invite users by sending an invitation to a distribution list.
4. Select the **Send** button to send the invitations. Consider the following:
  - Each invitation can only be used to create a user under the email address that it was sent to, and can only be used once.
  - Each invitation will expire seven days after the day it was sent.
  - Your user license count will not be affected until users accept the invitation and their accounts are created.

### Create users automatically

#### Use a mail handler, connect to an internal directory, or enable public signup

There are a few other ways to create a user in Jira. The following methods are more specialized and can fill a specific need of your team.

##### Automatically create a user via a mail handler

You can use a mail handler to allow Jira applications to create issues or comments via received email. The handler can also be configured to create new users based on the sender's email address.

This method can be used from time to time if you want Jira to create new user accounts from any received email messages whose **From:** field contains an address that does not match one associated with an existing Jira user account. This allows the creator of the email message to be notified of subsequent updates to the issue.

See [Creating issues and comments from email](#) for detailed information.

##### Connect to an internal directory with LDAP authentication

You can connect your Jira application to an LDAP directory for delegated authentication. This means that Jira will have an internal directory that uses LDAP for authentication only. Choose this option if you want to set up a user and group configuration within your application that suits your needs, while checking your users' passwords against the corporate LDAP directory. This option also helps to avoid the performance issues that may result from downloading large numbers of groups from LDAP.

See [Connecting to an internal directory with LDAP authentication](#) for more information on configuration.

### Allow users to sign up publicly

For some organizations using Jira Service Management, it's appropriate to allow users to create their own accounts without needing a Jira admin. This is a good way to empower users without using up all of your Jira Service Management licenses. However, this can raise some security concerns.

See [Enabling public signup and CAPTCHA](#) for more information about enabling public sign up and CAPTCHA.

### Select default applications for new users

If you have more than one Jira application, it's possible to select which applications new users will automatically be assigned to. If you manually create a user, the applications you select as defaults will be preselected. However, it's possible to change this while creating the user. If you allow users to sign up via email, via public signup, or an email handler, they will be given access to the applications you select.

1. In the upper-right corner of the screen, select **Administration**  **User Management**.
2. On the left-side navigation panel, select **Application access**.
3. Select **Set defaults for new users**.
4. Choose one or more applications that you want to set as default and select **Set defaults**.

You've now set the default applications to be used for new user creation. These users will be assigned to the default groups of the applications you have selected.

### Editing users

Modifying user information, such as name, email, address, and password, is easy with the Jira internal directory. If you are using an external authentication method such as LDAP or Active Directory, you'll have to make changes in that system rather than in Jira. See [Create users automatically](#) for more information.

#### Edit a username, full name, or email address

If you're using the Jira internal directory to manage users, you can modify these three attributes together, in a few simple clicks.

 When updating a username, note that Jira cannot update external usernames for example, users that are coming from an LDAP server or Crowd instance. However, Jira can update Jira users stored in an "Internal Directory with LDAP Authentication."

1. In the upper-right corner of the screen, select **Administration**  **User Management**.
2. Find the user in the user list by using the filter form at the top of the page.
3. Select the **Full name** of a user you want to edit.
4. Navigate to the **Actions** menu and select **Set Password**.
5. Enter and confirm the new password, and then select the **Update** button to save your changes.

#### Change a password

Jira admins can change user passwords directly in Jira when using the internal directory. A password cannot be changed if users are managed from an LDAP server or Crowd instance.

1. In the upper-right corner of the screen, select **Administration**  **User Management**.
2. Find the user in the user list using the filter form at the top of the page.
3. Click on the **username**.

4. Choose **Actions > Set Password**.
5. Enter and confirm the new password, and click the **Update** button to finish.

### Add a property to a user

A Property is an extra piece of information about a user that you can store in Jira. A Property consists of a Key of your choice, like "phone number" or "location", plus a corresponding Value (eg. "987 654 3210", "level three"). User Properties do not have an effect in the project apart from adding additional information about the user. Apps, however, can frequently use this data.

1. In the upper-right corner of the screen, select **Administration**  **> User Management**.
2. Find the user in the user list using the filter form at the top of the page.
3. Select the **Full name** of a user you want to edit.
4. Select **Actions > Edit Properties**. The **Edit User Properties** screen will be displayed.
5. Enter the new **Key** and its **Value**, then select the **Add** button to finish.

### Deactivating users

Have a user that no longer needs access to Jira? Rather than deleting a user, we recommend that you deactivate their account. Deactivating a user's account will prevent it from being used, but it will preserve that user's history of activity. You can also [anonymize a user](#), which can be useful if somebody requests to have their personal data erased.

#### Deactivate a user

Jira administrators can deactivate a Jira user, which disables that user's access to Jira. This avoids the need for a Jira admin to delete the user's account from the system.

This feature is useful when a Jira user leaves the organization or changes departments because their Jira activity history is preserved in the system. If a user with a deactivated Jira account needs access again at some point in the future, their Jira user account can be easily reactivated.

1. In the upper-right corner of the screen, select **Administration**  **> User Management**.
2. In the user list, find the user that you want to deactivate
3. In the Actions menu select **Edit Details**.
4. Deselect the **Active** checkbox.
5. Select **Update** to confirm the change.
6. The user will now appear in the user list with a strikethrough their username and full name, and the text **(inactive)**.

#### Note

- To deactivate a [project or component](#) lead, assign other users as the relevant project or component leads first. These users cannot be deleted without first replacing their roles. An error message will appear asking you to assign another user first.
- If your Jira instance is configured to use an external Atlassian Crowd user directory, the user will be deactivated in Jira if they are deactivated in Crowd.
- Jira does not deactivate users who are configured and deactivated/disabled in an external Microsoft Active Directory or LDAP-based user directory, with the exception of Jira users configured with "delegated LDAP authentication".
- You can activate a deactivated users by following the same steps from the preceding instruction.

#### When you deactivate a user, they:

- Will no longer be able to sign in to Jira.
  - Will not count towards your Jira user license limit.
  - Can't be assigned issues or added as a watcher to issues whenever issues are created or edited.
- However:

- A user who was assigned, was watching, or had reported any issues in Jira before their account is deactivated, will still appear as the respective assignee, watcher, or reporter of those issues. This situation remains until another user is specified as the assignee or reporter, the deactivated user is removed as a watcher, or the account is reactivated.
- A user who voted on any issues in Jira before their account is deactivated will continue to appear as a voter on these issues.
- Will continue to appear on the Jira user interface with "(Inactive)" displayed after their name.
- Can still be used to filter issues in a Jira search query.
- Will not receive any [email notifications](#) from Jira, even if they continue to remain the assignee, reporter, or watcher of issues.

## Deleting users

We recommend deactivating a user instead of deleting one. Check the previous section for more information.

 If you do want to delete a user, you need to keep a few things in mind:

- You can't delete a user from within Jira if you're using External User Management. However, you can deactivate the user.
- You cannot delete a user from Jira if they've reported any issue, commented on any issue, or been assigned to any issue.
- The filters and dashboards of a user will be deleted when the user is deleted, even if the filters or dashboards are shared with other users.
- All issues that have been reported by or assigned to the user you are attempting to delete, are respectively hyperlinked to a list of the individual issues in the Issue Navigator.

1. In the upper-right corner of the screen, select **Administration**  **> User Management**.
2. In the user list, find the user that you want to delete. You can also use the filter form at the top of the page.
3. In the **Actions** column select **Delete user**. Jira will check if the selected user has reported any issue, commented on any issue, been assigned to any issue, etc. The relation between the user and other parts of the system may prevent the deletion of the user.
4. Take any actions required to disassociate the user with Jira. The error message will give you exact instructions. They may include:
  - Reassigning any issues currently assigned to the user.
  - Bulk editing the issues created by the user and changing the Reporter to another active user. If the user you're deleting created any issues that are closed now and that you don't want to reopen, you'll need to allow editing of closed issues.
  - Changing the owner of shared dashboards owned by the user you're deleting. For more details, see [Managing dashboards](#).
  - Changing the project lead for any projects where the user is a lead.
5. If there are no issues assigned to, commented by, or reported by the user, the confirmation screen will display a **Delete** button. Select the button to delete the user.

## Anonymizing users

Anonymizing a user will hide or delete any data that can identify them as a real person. Any occurrences of their personal data, like username or full name, will be changed into an unrecognizable alias.

You can also anonymize a user when they request to erase their personal data from the system.

1. In the upper-right corner of the screen, select **Administration**  **> User Management**.
2. In the user list, find the user that you want to anonymize. You can also use the filter form at the top of the page.
3. In the **Actions...** column select **Anonymize user**. You'll be redirected to a page with details.

For more information, see [Anonymizing users](#).

# Assign users to groups, project roles, and applications

When a user is created in Jira, they'll be automatically added to the default user group for your installation (jira-users, jira-software-users, jira-servicedesk-users). As an administrator, you can choose to create a more granular security model by creating multiple user groups that grant different levels of access within the Jira instance (see more about user groups in the [managing groups](#) section).

 Managing 500+ users across Atlassian products?  
Find out how easy, scalable and effective it can be with Crowd!  
See [centralized user management](#).

This section will give you instructions on how to assign permissions by adding users to groups and assigning a user to a project role.

## Before you begin

You must have the Jira Administrator or Jira System Administrator global permission to be able to manage users in Jira applications.

## Add a user to a group

The best way to give a user access to specific Jira functions is to add a user to a predefined user group.

1. Select **Administration**  > **User Management** to view the user list.
2. Find the user in the user list using the filter form at the top of the page.
3. Select **Groups** in the Operations column.
4. Use the search box to find the group that you want to add the user to. You can add more than one group at a time in that search field if you need to add the user to multiple groups.
5. Select **Join selected groups** and the user will be added.

## Assign a project role

One way to give users access to a project role is to grant access at the user level. If you have fewer than 50 Jira users, you can manage user permissions by manually assigning users to a project role. If you have more than 50 users, we recommend adding users to a group that can then be assigned to a project role, as explained above.

1. Select **Administration**  > **User Management** to view the user list.
2. Find the user in the user list using the filter form at the top of the page.
3. Select **Project Roles** in the Operations column.
4. Select **Edit project roles** to add / remove a user from a project role. On this screen, you can also see the Groups that provide access to each project role.
5. Check the box for the project role that you want to give access to (**Administrators, Developers, or Users**) and select **Save** to finish.

## Assign a user to an application

You may need to give an existing user access to an application, or remove access to an application. Application access can be assigned and removed in the User browser. When you assign a user to an application in this manner, they will be added to that application's [default group](#). When you remove application access, the user is removed from all the groups that could grant them access to that application.

1. Open the **User** browser.
2. Select the user you wish to assign or remove access from. The user is displayed and the applications they are assigned to are displayed as checked under **Applications and groups**.
3. Check the box for the application/s you want to assign to a user. Unselect the box to remove access. Note that the changes are made in real time, as you add or remove access, the group memberships change.



# Monitor a user's activity

As a Jira administrator, you can view individual user activity or user sessions on a global level.

## View individual user login activity

 For all of the following procedures, you must be logged in as a user with the **Jira Administrators** [global permission](#).

Administrators may want to check individual user login activity to:

- See if certain users are still active in Jira.
  - View the age of a disabled user to clean up any ID's that have passed a certain timeframe.
1. Choose **Administration**  **> User Management**.
  2. Click on the **username** in the list.
  3. You will be presented with a window including;
    - Login count
    - Last login
    - Previous login
    - Last failed login
    - Current failed login count
    - Total failed login count
  4. Use this information wisely.

## View global user sessions

 For all of the following procedures, you must be logged in as a user with the **Jira System Administrators** [global permission](#).

Jira provides a list of users who are currently accessing Jira. Use this information to:

- Know who to contact before planned downtime.
  - Regularly monitor the average number of active users to evaluating your licensing quota.
  - View user count if the system is under duress.
  - And more!
1. Choose **Administration**  **> System**.
  2. In the left panel, select **Security > User Session** to open the Current User Sessions in Jira page.

### Note

It's possible to have session ID's for computers that are not logged in. For example, when someone accesses Jira without logging in, a unique session is created without a username. This is shown as "Not Available" in the User column.

# Manage password security

Create a more secure Jira environment by enabling a password policy, setting custom password settings, or enabling password similarity checks.

## Enabling a password policy

The Jira password policy is disabled by default. This policy is only useful when Jira users are able to change their own passwords. If Jira is connected to an external user management system (LDAP, Active Directory, Crowd), this policy should not be used since passwords are maintained externally from Jira.

1. Select **Administration** (⚙️) > **Security**.
2. Click **Password Policy** in the left panel. Select one of the following options:
  - a. **Disabled** The equivalent of having no password policy (this is the default).
  - b. **Basic** Requires passwords to be at least 8 characters long and use at least 2 character types. Rejects passwords that are very similar to the previous password or the user's public information.
  - c. **Secure** Requires passwords to be at least 10 characters long and use at least 3 character types including at least 1 special character. Rejects passwords that are even slightly similar to the previous password or the user's public information.
  - d. **Custom** Lets you use your own settings (see below for more information).
3. Click the **Update** button to finish.

## Setting custom password policies

There are many optional fields that can be set when you choose a custom password policy.

### Set 'Custom' password settings

Update the necessary fields to meet your company's password standards:

1. **Password Length** Set a minimum and maximum length for your passwords. The defaults are 8 and 255.
2. **Character Variety** Use these fields to set requirements around types of characters: uppercase letters, lowercase letters, special characters, and so on.
3. **Similarity Checks** See the section below for details on this feature.

### Similarity checks for 'Custom' password settings

This is a system check to make sure that your users aren't creating a new password that is too similar to the current password, the user's name, or email address. It can be set to **Ignored**, **Lenient**, or **Strict**.

### What's the difference between Lenient and Strict?

- **Lenient** checks for obvious similarities, like reversing the `username` or moving the front letter to the end.
- **Strict** checks for more subtle variations, like mixing up the letters or adding just one new character. It also performs a character frequency analysis.

## Enabling CAPTCHA

If your Jira application server is accessible from outside your organization's firewall, and you have enabled signup, then you may want to also enable CAPTCHA. CAPTCHA helps ensure that only real humans (and not automated spam systems) can sign themselves up to Jira. When CAPTCHA is enabled, visitors will need to recognize a distorted picture of a word (see example below), and must type the word into a text field. This is easy for humans to do, but very difficult for computers. See '[Enabling public signup and CAPTCHA](#)' for more information about enabling this option.

## Password FAQ

**Question:**What is Character Variety and why should I use it?

**Answer:**Character variety refers to the different types of characters you can create on a keyboard: lowercase letters, uppercase letters, numbers, and special characters. Requiring different character types makes passwords harder to guess, but it might also make them harder to remember. Use your best judgment when setting these fields, keeping in mind your company's requirements as well as your user base.

**Question:**Does this policy affect existing passwords?

**Answer:**The policy is only enforced as passwords are changed; there is no way to detect whether or not existing passwords satisfy the policy or to force the users to update their passwords if the policy has been changed. As a workaround, you can use [this Crowd REST resource](#) to forcibly change the users' passwords to something they won't know, thereby requiring them to reset it to get back in, and the password reset enforces the policy rules.

# Prevent automatic login

## Overview

When a user logs in to Jira, they have the option of making Jira remember their login information by selecting the 'Remember my login' checkbox before they click the 'Log In' button. When they do that, a 'Remember my login' token is stored by the Jira server and a cookie containing this token is set in the user's browser.

A user who revisits Jira from the same computer and browser, will automatically be logged in if Jira detects that one of the user's 'Remember my login' tokens has a matching token contained in one of the browser's cookies. If the user logs out of Jira, the 'Remember my login' token is cleared from the Jira server.

To maximize and maintain the security of your Jira instance, Jira provides features for:

- Disabling 'remember my login' functionality for the Jira instance.
- Clearing 'Remember my login' tokens for individual user accounts.
- Clearing all 'Remember my login' tokens stored by your Jira instance.

Manage automatic logins:

- To maximize security by requiring a user to enter all of their credentials each login.
- If users have been accessing your Jira application in a public environment.
- If users aren't in the habit of formally logging out of Jira.

### Before you begin

You must be logged in as a user with the **Jira Administrators** [global permission](#) to complete any of the following procedures.

## Clear a 'remember my login' token for a specific user

Jira administrators can clear all 'Remember my login' tokens associated with a user's account through the Jira administration console.

1. Choose **Administration**  **> User Management**.
2. Find the user in the list and click the **Username** or **Email Address** of the user whose 'Remember my login' tokens you wish to remove. Details about that user and their login information is displayed.
3. Click the **'Remember My Login'** link to display that user's **Remember My Login** page.
4. Click the **'Clear All'** button to remove all 'Remember my login' tokens associated with this user account from the Jira server.

## Clear all 'remember my login' tokens for the entire Jira instance

Jira administrators can also clear all 'Remember my login' tokens from their Jira instance with a few simple clicks.

1. Choose **Administration**  **> System**.
2. In the left panel, select **Security > Remember My Login** to open the Remember My Login for All Users page.
3. Click the **'Clear All'** button to remove all 'Remember my login' tokens from the Jira server.

## Disable 'remember my login on this computer' option for your Jira instance

If you never want Jira to remember login tokens, you can choose to disable 'remember my login' tokens for the entire Jira instance.

**Option 1 (recommended)**

The checkbox for this option can be disabled by setting the `jira.option.allowcookiesproperty` to `false` in your `jira-config.properties` file. You will need to restart Jira in order for this change to take effect.

**Option 2**

Edit the `./atlassian-jira/includes/loginform.jsp` file.

# SAML SSO for Jira Data Center applications

Security Assertion Markup Language (SAML) is an XML-based data format that allows a service to exchange authorization data with an identity provider (IdP). The most common use case is allowing a user to sign in to multiple software applications using the same authentication details, usually a username and password. This is referred to as single sign-on (SSO).

Atlassian provides the [SSO for Atlassian Server and Data Center](#) app that allows Jira Server and Data Center applications to connect to your IdP so that you can provide your users with an SSO experience.

**i** This page describes the latest SSO features available in Jira Software Data Center and Jira Service Management Data Center applications. For earlier versions or Server products, the functionality might be limited. Check if you can upgrade your SSO app, or find the SSO functionality under SSO 2.0 in Administration.

[SSO for Atlassian Server and Data Center](#) only handles authentication. Application access and any required authorizations, such as ensuring that users belong to the appropriate groups/roles and have the necessary permissions, should be configured in the user directory and/or the application itself.

## **i** Looking for a cross-domain SSO solution?

Crowd Data Center 3.4 with its Crowd SSO 2.0 feature offers one solution for authenticating to Server and Data Center applications, and setting it up takes only minutes. Are you ready for the change? See [Crowd SSO 2.0](#).

## Setting up single sign-on

You'll need to configure your application and your IdP to provide single sign-on for your users.

### Supported identity providers

SAML single sign-on should work with any identity provider implementing the SAML 2.0 Web Browser SSO Profile, using the HTTP POST binding.

We currently perform tests with the following identity providers:

- [Microsoft Active Directory \(using ADFS 3.0\)](#)
- [Microsoft Azure Active Directory](#)
- [OneLogin](#)
- [Okta](#)
- [PingIdentity](#)

### Setting up SSL/TLS

To make sure that SAML authentication is secure and private, you need to set up SSL/TLS in the application. You can find the relevant steps in our [Running Jira applications over SSL or HTTPS](#) page.

Once set up, you need to make sure that the application's [configured base URL](#) is using the HTTPS protocol.

Setting up SSL/TLS using a reverse proxy

If you want to use a reverse proxy, please refer to these product-specific documents, describing the exact configuration steps:

- [Proxying Atlassian server applications with Apache HTTP Server \(mod\\_proxy\\_http\)](#)
- [Integrating Jira with Apache using SSL](#)
- [Securing your Atlassian applications with Apache using SSL](#)

When using a reverse proxy that terminates SSL/TLS, you need to make sure that the request URL the application server sees matches the fully-qualified domain name for the reverse proxy. This is usually achieved by configuring the `<Connector>` directive with the appropriate `proxyName`, `proxyPort`, `secure`, and `scheme` settings. Please check the documentation above for specific examples.

## Setting up your identity provider

If you want your application to provide SSO, you'll need to add it to your IdP. The exact process varies depending on the IdP, but you'll usually need to:

- Define an "application" in your IdP.
- Provide some data about the application, including data you can access on your application's Authentication methods screen.
- Make sure the NameID attribute of the users in your IdP is set to the username in your Atlassian application.
- Give the appropriate users permission to use the application.

At the end of the setup process, your IdP will provide you with a set of data that you'll need to configure your Atlassian application.

## Configuring SAML SSO authentication in your Atlassian application

1. Open the Authentication methods screen by selecting **Administration**  > **System** > **Authentication methods**.
2. Select **Add configuration**.
3. In the **Authentication method** menu, select **SAML single sign-on**.
4. Create a unique name for your configuration.
5. Configure the following **SAML SSO settings**:

Setting	Details
Single sign-on issuer	This value is provided by your IdP, as part of setting up SAML. It's sometimes also called "Entity ID".  The issuer is the IdP your application will be accepting authentication requests from.
Identity provider single sign-on URL	This value is provided by your IdP, as part of setting up SAML.  It defines the URL your users will be redirected when logging in.
X.509 Certificate	This value is provided by your IdP, as part of setting up SAML. This is sometimes referred to as a "Signing certificate". The key usually starts with "-----BEGIN CERTIFICATE-----".  This contains the public key we'll use to verify that all received SAML authentication requests have been issued by your IdP.
Remember user logins	If selected, successful user logins will be remembered in the user's browser. When browsing their application, users will be logged in automatically without having to authenticate again using SAML.
Username mapping	An attribute from your IdP that uniquely identifies a user and can be mapped to the username in Jira. For example, if you entered <code>\${NameID}</code> , we would use the values of this attribute from your IdP as usernames. Check your IdP docs for the list of attributes.

6. The following information is provided on the Authentication methods screen and will be required to configure your IdP. Copy and paste these links to your identity provider:

Setting	Details
Assertion Consumer Service URL	This is the URL the IdP will return SAML authentication requests to.
Audience URL (Entity ID)	This is the URL the IdP will prepare SAML authentication requests for.

7. The following settings are optional and can be configured as you see fit:

Setting	Details
JIT provisioning	If <b>Create users on login to the application</b> is selected, users will be created and updated automatically when they log in through SSO to Atlassian Data Center applications. See <a href="#">JIT user provisioning</a> for details.
SAMLSSO behavior	If <b>Remember user logins</b> is selected, users will be logged in automatically as their login information will be remembered.
Include customer logins (Jira Service Management only)	If selected, all login requests from your Jira Service Management customers using the customer portal will be redirected to the configured IdP. If not selected, customers will have to log in through the customer portal.
Login page settings	If <b>Show IdP on the login page</b> is selected, users will see identity provider on login page.

8. Select **Save configuration**.

Once you've configured both your application and your IdP, you're ready to start using SSO.

### Best practices

- To ensure the security and privacy of your authentication process, make sure that both your Atlassian application and your IdP are using the HTTPS protocol to communicate with each other, and that the configured application base URL is the HTTPS one.
- SAML authentication requests are only valid for a limited time. You should make sure the clocks on the server running your application/s and the IdP are synchronized.
- If users and groups in your application are configured using [User Directories](#), you'll usually want to use the same LDAP directory to be the source of users for both your IdP and Atlassian application. Users need to exist in the user directory *before* they can log in using SSO.

### Troubleshooting

- If you make a mistake configuring the SAML authentication, or are unable to log in using your IdP, you can bypass SAML authentication by using the `auth_fallback` functionality. See [Bypass SAML authentication for Jira Data Center](#) for details.
- If an authentication error occurs, the user will only see basic details about what went wrong. For security reasons, the details about the underlying problem are not shown. You'll need to check the application logs to see the cause of the problem.
- In some cases, you might also experience errors shown by your IdP. For those, you will need to use the support and tools provided by your IdP, rather than Atlassian support.
- If you use **SAML as primary authentication** and have CAPTCHA enabled in the application, users that use HTTP basic authentication (for example in REST resource calls) may get locked out if they enter an incorrect password too many times. In these cases, an administrator will need to reset the user's CAPTCHA in the user list screen.

# Anonymizing users

You can anonymize users in Jira to hide or delete any data that can identify them. Anonymization helps you stay compliant with General Data Protection Regulation (GDPR) and the right to be forgotten, and is most often needed when somebody is leaving your organization and requests to have their personal data erased.

**Compatible applications.** When anonymizing users, we'll change or erase their personal data in Jira Core, Jira Software, Jira Service Management, and Portfolio for Jira.

- On this page:**
- [What does the anonymization involve?](#)
  - [Anonymizing a user](#)
  - [Understanding the scope of anonymization](#)
  - [Anonymization limitations](#)
  - [Previous limitations](#)
  - [Troubleshooting](#)
  - [Known issues](#)
  - [APIs](#)
  - [For app developers](#)

## What does the anonymization involve?

Every user in Jira is associated with some items they might have an issue assigned, be referenced in permission schemes, or mentioned in comments by their teammates. Some pieces of user data are anonymized, while others are completely erased. You can find a full list of those items in the following sections and in Jira when you start anonymizing a user.

The two main things to understand for anonymized users are how we treat their:

- **Username:** Changed into an anonymous, unrecognizable alias, like *jirauser80900*.
- **User profile:** Completely anonymized and looks like a new user profile. The full name, which is often displayed around Jira, is given an anonymous alias. For example, *user-ca31a*.

The following are examples of a user **Friendly Robot** (username: *friendlyrobot*) who has been anonymized and is now **user-ca31a** (username: *jirauser80900*).

Example	Before	After
Issue reporter		
Comment		



### Anonymizing a user

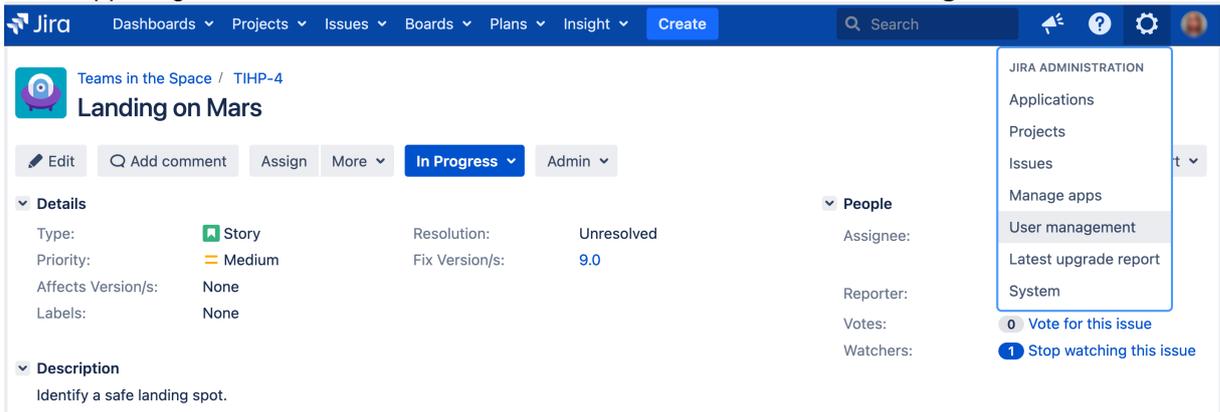
**i** For all of the following procedures, you must be logged in as a user with the **Jira administrator** permissions. For details, see [Permissions overview](#).

You can anonymize users in two ways. The method you use depends on whether the user is still active, or has been deleted.

Whichever option you choose, you will be redirected to a separate **Anonymize user** page that shows details about the selected user and lists all associated items that will be transferred, anonymized, or deleted. Your user won't be anonymized yet, so feel free to try it.

### Anonymize active users

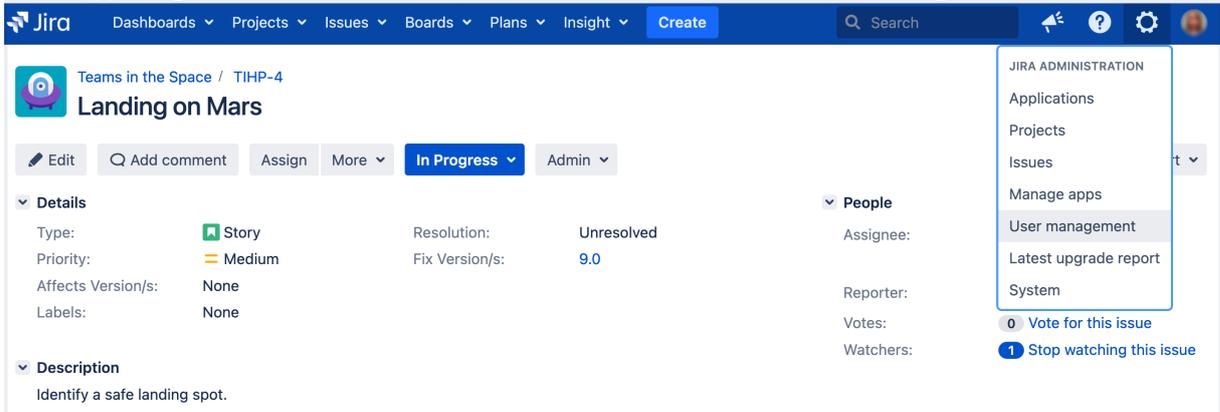
1. In the upper-right corner of the screen, select **Administration** **> User management**.



2. In the User browser, find the user you want to anonymize, and then select **... > Anonymize user**.

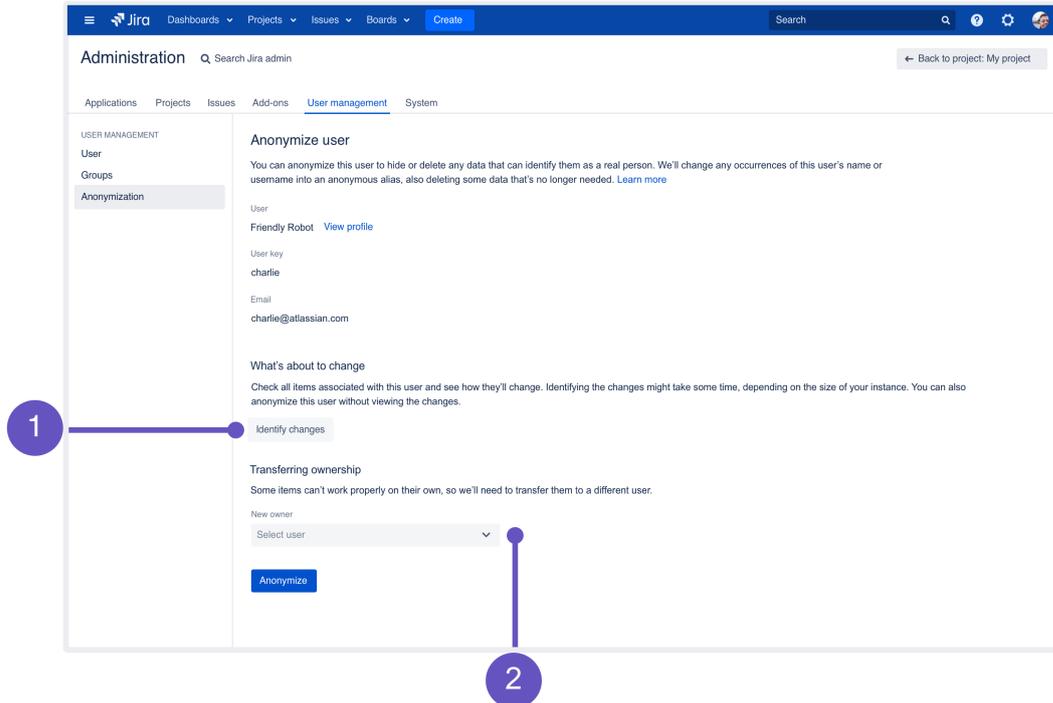
### Anonymize deleted users

1. In the upper-right corner of the screen, select **Administration** **> User management**.



2. Enter the username and select **Anonymize**. When anonymizing users that have been deleted, you'll see a **DELETED** label next to their username.

### Understanding the scope of anonymization



- 1. Identify changes:** You can click this button to search Jira for any items associated with a user and have them displayed here. This is optional, we will anonymize all of these items even if you don't view them. You can see the complete list of items in [Whats about to change after anonymization](#).
- 2. Transferring ownership:** Some items owned by a user, like Project Lead or Component Lead, might break things if left without the owner. You'll need to select a new owner here, and we'll transfer the items for you. This section won't be displayed if there's nothing to transfer.

## What's about to change after anonymization

If you choose to display items associated with a user, they will typically be displayed in four sections:

- [Transferred items](#)
- [Anonymized items](#)
- [Deleted items](#)
- [Actions required on your side](#)

✔ If Jira doesn't find the user's data in any of the sections, they won't be displayed at all. In that case, what you see may differ slightly from these examples.

### Transferred items

Some items won't work properly with inactive users, so you'll need to choose a new owner for these items. For example, an inactive Component Lead might break the Default assignee option. You can choose any user with proper permissions, but it's probably best to transfer them to a project admin or somebody who has taken over the tasks of the anonymized user.

- Project lead
- Component lead
- Filter subscriptions
- + *Custom items added by Marketplace apps*

### Anonymized items

Anonymized data includes items with any occurrences of the users name or username. As mentioned earlier, we change these occurrences into an anonymous alias generated specifically for this user. The items themselves need to remain in Jira as they affect other areas or users these are usually comments, work logs, workflows, and so on.

- User profile (anonymizing user data, such as email, name, display name, removing avatars, remember me tokens, user settings, and browsing history)
- Workflows
- Draft workflows
- User key entries in the database
- Comments
- Work logs
- Audit log
- Board owners
- Board admins
- Card colors
- Notifications (recipients)
- Notifications (events)
- Jira invitation emails
- Atlassian Notifications messages
- Atlassian Troubleshooting and Support Tools app
- Webhooks
- Jira activity stream
- Hipchat app
- + *Custom items added by Marketplace apps*

#### Deleted items

These items are specific to a user and don't affect anybody else, so there's no point in keeping them in Jira. These can be associations in various schemes (don't worry, we won't delete the schemes), personal filter subscriptions, or personal roles the ones used only by this user. Once you anonymize the user, these will be gone forever.

- Personal project roles
- Personal filter subscriptions
- Occurrences in notification schemes
- Occurrences in permission schemes
- Permissions in shared filters and dashboards
- Atlassian Notifications user properties
- + *Custom items added by Marketplace apps*

#### Actions required on your side

Finally, there are items which we can't anonymize, and you'll need to change them manually. This section lists various items that include JQL queries with users personal data or data stored in 3rd party apps.

#### Anonymization limitations

 Because of the following limitations, some personal data will not be anonymized. You can start anonymizing users, and then complete anonymizing the missing data once we release the fixes. To complete anonymizing these items later, you'll need to [retry the anonymization](#), which will anonymize only items that haven't been anonymized before.

#### External user directories

Jira can't anonymize users that are stored in external user directories (like Crowd). You need to remove a user from the external directory, sync the directory with Jira, and only then anonymize them.

To view your user directories:

1. In the upper-right corner of the screen select **Administration**  **User management**.
2. On the left-side panel, select **User directories**.

## Some user keys in Insight - Asset Management

[Insight - Asset Management](#) is an app bundled with Jira Service Management Data Center 4.15 and later. Its also available for earlier Data Center versions.

In some cases, when you anonymize users, their user keys wont be anonymized in Insight - Asset Management. This affects user keys of users created before Jira Service Management 4.7 (or Jira Core and Jira Software 8.7). In these versions, user keys look the same as usernames, so some personal data might be visible. Users created in later versions will be fully anonymized in Insight - Asset Management as well. The reason for this difference is that later versions use a different pattern for usernames and user keys, and in this case, were only able to anonymize the new pattern.

### JQL queries

Personal data that appears in JQL queries wont be anonymized. Queries that are specific to Jira Service Desk will be shown in the **Actions required on your sidelist**, so it should be easy to edit them, but all the remaining ones wont be included. You'll need to review all JQL queries and manually delete usernames if they appear inside.

### Full names in issue history for recent users

When anonymizing users, most of the data that appears in the issue history will be anonymized. However, full names won't be anonymized for users who were created in Jira 8.4 or later, or who were created earlier and anonymized.

That's because Jira 8.4 changed the format of user keys. Full names for users created before this version will be anonymized.

Related issue: [JRASERVER-71153](#) - Usernames not fully anonymized in issue history CLOSED

### Third-party apps

Personal data stored in 3rd party apps won't be anonymized by default. However, weve created extension points app vendors can use to be notified when a user is being anonymized and to anonymize the related data. To check if an app supports the anonymization, contact the vendor directly or check their documentation.

### Previous limitations

The following limitations have already been fixed.

This limitation has been FIXED

You can't anonymize users that have been deleted although information on the pages in Jira might suggest otherwise. We've planned to add this feature since the beginning but had to descope it eventually.

This limitation has been FIXED

Personal data might still appear in the issue history, which shows all past activity on an issue. For example, if an issue was reassigned from one user to another, both these users' original names will be shown in the history even if you anonymized them.

This limitation has been FIXED

If a user has been set as a default value of a **text** custom field, this value won't be anonymized. You'll need to review your custom fields and change the default value manually.

This limitation has been FIXED

If a user is mentioned in the project description (**Project settings > Details**), this mention won't be anonymized. That's not a typical place where you'd mention a user, but be aware that this can happen.

## Troubleshooting

If anonymization fails, user data may be partially anonymized. If you encounter this problem, you can use the audit log to find the partially anonymized user, and retry their anonymization.

For more information, see [Retrying anonymization](#).

## Known issues

Check out the following article for more details: [An app was disabled when anonymizing a user](#).

## APIs

You can also anonymize your users by using the API.

For more information, see [Anonymization API](#).

## For app developers

If you're an app developer, we have created extension points that will inform your app when an admin anonymizes a user in their Jira instance. This lets you take the appropriate steps to anonymize any user data stored in your app.

For more information, see [Developer docs: Anonymizing users](#).

# Retrying anonymization

The anonymization of a user in Jira is not something that can be reversed. Even if it failed or was interrupted, the user will be partially anonymized, so you won't be able to find and anonymize them like you did before. Here, you can learn how to find the anonymized user and retry the anonymization.

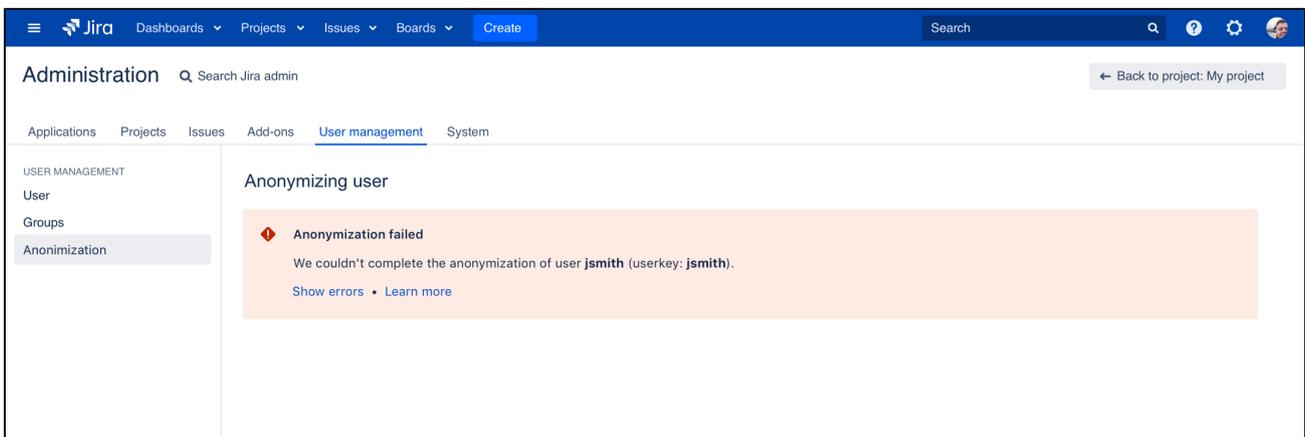
## Skip to

- [Am I in the right place?](#)
- [Fixing the problems](#)
- [Retrying the anonymization](#)

## Am I in the right place?

You will need to complete the anonymization in two cases:

- User anonymization finished with an error.
- Jira went down while anonymizing a user, and you're not sure whether it completed or not.



## Fixing the problems

Before retrying the anonymization, you need to fix the problems that caused it to fail in the first place.

## Logs

If the error message isn't enough, you can find more details in the [Jira application log](#):

```
<home-directory>/atlassian-jira.log
```

## Retrying the anonymization

To retry the anonymization of a user, you'll need to provide the following details:

- Current username (might be the same as original, or already anonymized)
- Original username
- Original user key

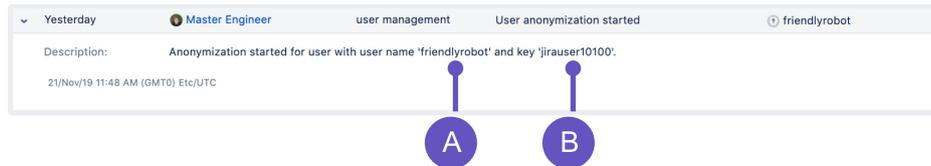
## Getting user details from the audit log

1. Go to **Administration > System > Audit log**.

2. Look for an entry Users key changed. If this entry exists, it means that the original username has already been anonymized. If it doesn't, the user still has the original username, so you can use it as *current username*.



- A. Anonymized full name. Click it to open the user profile and see the current username. It will most likely be the same as the user key that you can see on this image.
3. Look for an entry User anonymization started. This entry contains the original username and user key.



A. Original username.

B. Original user key.

### Retrying the anonymization

1. Go to **Administration > User management > Anonymization**.
2. Click **Retry anonymization** at the top-right.
3. Enter the current username, original username, and original user key.
4. Click **Anonymize**. The anonymization will look just the same as when you ran it for the first time, but will omit things that have already been anonymized.

# Enabling public signup and CAPTCHA

Enabling public signup allows users to create their own accounts. If it's not enabled, then only a Jira administrator can [create new user accounts](#).

If users create accounts that gives them access to Jira Core, Jira Software, or the agent view of Jira Service Management, these accounts will consume a license for these applications. If you enable public signup only for the Jira Service Management customer portals and customers create their own accounts, these accounts do not consume a license, as they have limited access to Jira.

 For security reasons, even if you enable signup, it is still necessary for users to have the appropriate [project permissions](#) before they can see or create issues. Note that you can use [automatic group membership](#) to add all new users to appropriate groups.

Before you begin

## Customer signup in Jira Service Management

If you're using Jira Service Management, it's worth noting that it has two public signup forms:

- **Agent (user) signup:** This is a regular signup for users who consume licenses and have access to Jira. It's the same signup that Jira Core and Jira Software users use. It can be accessed from the main Jira login page.
- **Customer signup:** This is a signup for external customers who don't consume licenses and have access only to the project's customer portal where they can raise requests. This signup doesn't require Jira to be in the public mode, as opposed to the regular signup.

## Enabling CAPTCHA

- If your Jira application server is accessible from outside your organization's firewall, and you have enabled signup, then you may want to also enable **CAPTCHA**. CAPTCHA helps ensure that only real humans (and not automated spam systems) can sign themselves up to Jira.

## Enabling public signup for Jira Core, Jira Software, and the agent view in Jira Service Management

1. Log in as a user with the **'Jira Administrators'** [global permission](#).
2. Choose **Administration** () > **System**. Select **General Configuration** to open the Administration page.
3. Click **'Edit Configuration'** at the end of the page.
4. In the 'Mode' drop-down, select **'Public'**.
5. Click the **'Update'** button at the bottom of the screen.
6. Log out of Jira, then click the **'Log In'** link at the top right of the screen and verify that the **'Sign Up'** link is displayed at the bottom of the login screen.

## Enabling public signup for Jira Service Management customer portals

With public signup enabled, agents can invite new customers to a service project, and new customers can create accounts on the Customer Portal and through email. Enabling public signup for your service project also enables a honeypot technique which helps prevent spambots from creating accounts through the customer portal.

## Enabling public signup

You must first enable public signup at the system level:

1. Log in as a user with the **'Jira Administrators'** [global permission](#).
2. Choose **Administration** () > **Applications**. Scroll down to the **Jira Service Management** section and choose **Configuration**.

3. In the **Public signup** section, allow project admins to enable public signup.

### Enabling verification emails

After enabling public signup, we recommend that you also enable verification emails. This adds security to your Jira instance and makes sure that all customers are exactly who they say they are. This option should be enabled by default unless you haven't configured outgoing email.

1. Enable and configure outgoing email so Jira can send verification emails. For more info, see [Configuring an SMTP mail server](#).
2. In the **Public signup** section, enable verification emails.

### Opening a service project

You or a service project administrator can then open a service project at the project level:

1. Go to **Project administration > Customer permissions**.
2. Select **Anyone can email the service desk or raise a request in the portal**.

New customers will be added to the **Service Desk Customers** project role. Note that customer accounts created via public signup don't count towards a Jira Service Management license.

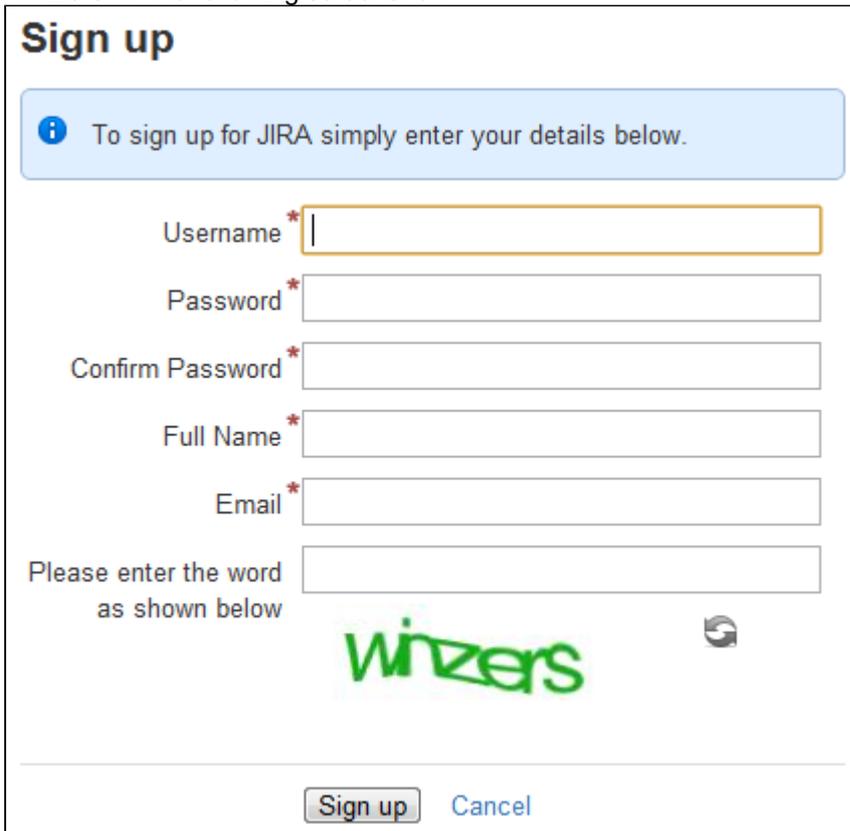
 In situations where users are unable to change their passwords, check that a Delegated Authentication Directory is not the highest in the order of User Directories. As a workaround, you can change the order of User Directories, or alternatively use a connection to a LDAP directory instead.

### Enabling CAPTCHA for Jira application login screens

CAPTCHA can be enabled so that anyone attempting to sign up to your Jira instance through the Jira login screen will be presented with a random sequence of letters, that they must type to confirm they're a real person. This is to try prevent spamming, and malicious attacks.

1. Log in as a user with the **'Jira Administrators'** [global permission](#).
2. Choose **Administration** () **> System**. Select **General Configuration** to open the Administration page.
3. Click **'Edit Configuration'** at the end of the page.
4. Locate 'CAPTCHA on signup' and select **'On'**.
5. Click the **'Update'** button at the bottom of the screen.

6. Log out of Jira, click the **Log In** link at the top right of the screen, then click the **Sign Up** link and verify that a random sequence of letters is displayed at the bottom of the **Sign Up** screen e.g. "winzers" in the following screenshot:



**Sign up**

**i** To sign up for JIRA simply enter your details below.

Username \*

Password \*

Confirm Password \*

Full Name \*

Email \*

Please enter the word as shown below

winzers 

[Cancel](#)

# Managing groups

## Jira groups

A Jira group is a convenient way to manage a collection of users. You can use groups throughout Jira to:

- Allow application access.
- Grant global permissions or project specific access.
- Receive email notifications.
- Access issue filters and dashboards.
- Reference workflow conditions.
- Integrate with project roles.

**i** Managing 500+ users across Atlassian products? Find out how easy, scalable and effective it can be with Crowd! See [centralized user management](#).

## Jira default groups

Two groups are automatically created when you install Jira for the first time: the jira-administrators group and one user group associated with the application.

Group	Application	Description	Use
<b>jira-administrators</b>	All	Contains people who are Jira system administrators. By default, this group: <ul style="list-style-type: none"><li>• is a member of the <b>Administrators</b> project role.</li><li>• has the <b>Jira Administrators</b> and the <b>Jira System Administrators</b> global permissions.</li></ul>	<ul style="list-style-type: none"><li>• Membership should be limited to a few Jira Administrators or super users.</li><li>• Provides unlimited access.</li><li>• Recommended to never delete or alter permissions for this group because it would limit accessibility to the Jira instance.</li></ul>
<b>jira-core-users</b>	Jira Core	By default, these groups have the <b>Browse Users</b> , <b>Create Shared Filter</b> , <b>Bulk Change</b> and <b>Manage Group Filter Subscriptions</b> global permissions.	<ul style="list-style-type: none"><li>• Optional user groups.</li><li>• May be useful if your Jira instance has very few users that require generic, standard access.</li><li>• Can be deleted if your Jira instance requires granular, specific access for individual groups of users.</li></ul>
<b>jira-software-users</b>	Jira Software		
<b>jira-service-desk-users</b>	Jira Service Management		

**i** Note

If you're using External User Management, you won't be able to create, delete, or edit groups or group membership from within JIRA, and automatic group membership will not apply. However, you'll still be able to assign groups to [project roles](#).

# View, create, or delete a group

## Before you begin

You must be logged in as a user with the **Jira Administrators** or **Jira System Administrators** global permission to perform the following procedures.

## View the group browser

The Group Browser in Jira allows you to view, create, and edit groups, while also allowing you to modify members, and view group permissions and settings.

1. Choose **Administration** () > **User Management**.
2. Select **Groups** to open the Group Browser.
3. Click the group name to see the permissions, email notification schemes, security levels, and saved filters.

## Create a group

Create new groups in Jira to customize security permissions based on roles. Users may be added to many groups depending on the level of access that they need to do their job.

1. From the group browser, type the new group name in the **Add Group** form.
2. Click **Add Group** and you're done.

## Note

New groups are created without access to Jira functions so you'll have to assign permissions to the group before members can inherit functionality.

## Delete a group

## Before you delete

- Check whether the group is being used by any [permission schemes](#), [email notification schemes](#), [issue security levels](#), or saved filters.
- Consider the impact this may have on users in that group. For example, if a user receives access to a specific feature only from this group, then the user will no longer have that permission and it may impact their work.

1. Choose **Administration** () > **User Management**.
2. Select **Groups** in the left panel to open the Group Browser.
3. Click **Delete** in the Operations column .
  - a. You will be redirected to a confirmation screen that explains that users will be removed from the group through its deletion. The users themselves will not be deleted from Jira during this operation.
  - b. Note that you cannot delete a group that is currently the only default group for an application. The **Delete** link will be greyed out.
4. Consider carefully and click **Delete** to finish (or **Cancel** if you've decided to reconsider).

# Modify group membership

## Editing group membership

To edit a group's membership, click the **Edit Members** link in the row for that group in the **Group Browser**. This takes you to a form that allows you to add or remove users from the group.

### Note

- When a user is created and assigned to an application, they are automatically added to that application's Default group.
- If you have a user limited license (e.g. personal license) and have reached your user limit, you will not be able to assign any further users to groups with log in [permissions](#) (i.e. application access) without first reducing the number of users with log in [permissions](#).
- If the group has the 'Jira System Administrators' [global permission](#), you cannot edit its membership unless you have the 'Jira System Administrators' global permission.

## Automatic group membership

To automatically add newly-created users to a particular group, you can assign the group as an [application's default group](#), and then assign the application as the [default application for user creation](#). Or specify the group name in the 'Default Group Memberships' option when [Connecting to an LDAP directory](#). See [Adding users to groups automatically](#) for instructions.

# Assign group access to a project role

You can grant access to project roles and applications through groups. Simply add a user to a group with predefined security settings to give them the access they need. Creating a clear security model, with specialized user groups, that grant specific access to applications is the easiest way for longterm admin support.

The best way to give users access to a project role is to grant access to a group. This way you can assign a group access and then simply add a user to the group and save yourself some time managing individual user permissions.

1. Select **Administration** () > **Projects**.
2. Click the title of the project that you want to assign permissions to.
3. Click **Users and Roles**.
4. Click the **Add users to a role** link.
5. Type the group (or user) names you want to add to a role.
6. Select the role from the drop-down.
7. Click **Add** to finish.

See '[Assign users to groups, project roles, and applications](#)' for more information.

# Manage group access to applications

Users need to belong to a group to access Jira applications. Administrators can assign access to groups through the Application access page.

Each application that's licensed in Jira is listed on this page, and each application will display its total allowed users, and users remaining. Each application should have at least one group associated with it, and one default group should be selected. When a user is created in Jira, and assigned to an application, they will be automatically added to the default group for that application. We strongly recommend you only have one default group assigned to each application. Use the default group to allow application access, and [create and manage other groups](#) to control [project specific permissions and access](#).

For example, when Jira Software is installed, it automatically creates the `jira-software-user` group in Jira, and assigns it as the default group for the Jira Software application. When you create a user and select Jira Software as the application they should have access to, they will be automatically added to the `jira-software-user` group. Once Jira Software is installed, you can add further groups to the application on the Application access page, and all members of those groups will have access to Jira Software. If a user is a member of more than one of those groups, they will only consume one user on your Jira Software license. For more information on creating users, see [Create, edit, or remove a user](#).

 You need to have the [Jira Administrator global permission](#) to access and edit the Application access page.

## Before you get started, know that:

- Each application that's licensed in your Jira instance is listed on this page.
- Each application will display its total allowed users and users remaining.
- Each application should have at least one group associated with it, and one default group should be selected.

## Assign a group to an application

All members of the groups assigned to an application will be able to log in to that application. You may assign multiple groups to any application. The users of these groups will count towards the user tier of your license. If a user is a member of multiple groups assigned to an application, they will only count as one licensed user.

1. Choose **Administration**  **> Applications**.
2. Select **Application access** in the left-hand menu. The Application access page displays all your applications and their associated groups, including their default groups.
3. Locate the application you want to assign a specific group to, and click the **Select group...** drop-down.
4. Select the group you wish to add. The group will be added to the application.

### Assigning a group to multiple applications

You may assign the same group to several applications. One reason to do this is to ensure members of the group always have full application access. For example, you may have users who requires full access to all your applications. You could create a separate group for them, and add this group to each application. Care should be taken when assigning a group to multiple applications, as the group members will consume a license for each application. The only exception is Jira Core. A user with access to any other application automatically has access to Jira Core, so they will not consume a license for Jira Core if they belong to a group associated with another application.

## Assign a default group to an application

When an application is installed, it automatically creates and assigns a default group to itself. You can manually change the default group, and we strongly recommend that you only have one default group assigned to an application.

1. Choose **Administration**  > **Applications**.
2. Select **Application access** in the left-hand menu. The Application access page displays all your applications and their associated groups, including their default groups.
3. Check the box in the **Default groups** column for the group you want to assign as a default group. Note you **must** have at least one default group at any time. If you want to change the default group, you must first assign a second default group before you can un-check the box for the current group.

You can also set which application you'd like new users to be added to in Jira. This is covered in more detail on ['Assign users to groups, project roles, and applications'](#).

#### **Assigning multiple default groups to an application**

We strongly recommend you only have one default group for each application. New users created and assigned application access are added to that application's default group. If this group is also assigned to another application, the new user will also gain access to the additional application, potentially creating a security hole in your system.

For example, if you assign a group called Group A to Jira Software and make it the default group, all new users added to Jira Software will be added as members of Group A. If you then add Group A to Jira Service Desk, all users in Group A will now have full access to both Jira Software and Jira Service Desk. This also means that when you create a new user and add them to Jira Software, they will also gain access to Jira Service Desk and consume a license for both applications.

# Advanced user management

- [Allowing connections to Jira for user management](#)
- [Diagrams of possible configurations for user management](#)
- [Managing nested groups](#)
- [User management limitations and recommendations](#)

 Managing 500+ users across Atlassian products?  
Find out how easy, scalable and effective it can be with Crowd!  
See [centralized user management](#).

# Allowing connections to Jira for user management

You can allow other applications to connect to your Jira Server for management of users and groups, and for authentication (verification of a user's login).

Examples of such applications: Atlassian Confluence, FishEye/Crucible, Bamboo, or another Jira Server.

**Note:** For all of the following procedures, you must be logged in as a user with the **Jira Administrators** global permission.

## On this page:

- [Allowing an application to connect to Jira for user management](#)
- [Diagrams of some possible configurations](#)

**i** Managing 500+ users across Atlassian products? Find out how easy, scalable and effective it can be with Crowd! See [centralized user management](#).

## Allowing an application to connect to Jira for user management

Subject to certain limitations, you can connect a number of Atlassian applications to a single JIRA application for centralized user management.

**When to use this option:** You can connect to a server running **Jira 4.3** or later, **Jira Software 7.0** or later, **Jira Core 7.0** or later, or **Jira Service Management (formerly Jira Service Desk) 3.0** or later. Choose this option as an alternative to Atlassian Crowd, for simple configurations with a limited number of users.

### To configure an application to connect to Jira as a user server:

1. Add the application:
  - a. Choose **Administration** (⚙️) > **User Management**.
  - b. Select **Jira User Server**.
  - c. **Add** an application.
  - d. Enter the **application name** and **password** that the application will use when accessing your Jira server application.
  - e. Enter the **IP address** or addresses of the application.
  - f. **Save** the new application.
2. Set up the Jira user directory in the application:

(For example, see [Connecting Confluence to Jira applications for user management](#) or [Connecting Jira applications to another server](#).)

  - a. Log in to the application that is going to connect to your Jira server for user management.
  - b. Go to the application's '**User Directories**' administration area.
  - c. Add a new directory of type '**Atlassian Jira**'.
  - d. Define the directory order (see [Managing multiple directories](#)).
3. Create any groups in your Jira server that are required by the other application. For example, see [Connecting Confluence to Jira for User Management](#).

## Diagrams of some possible configurations

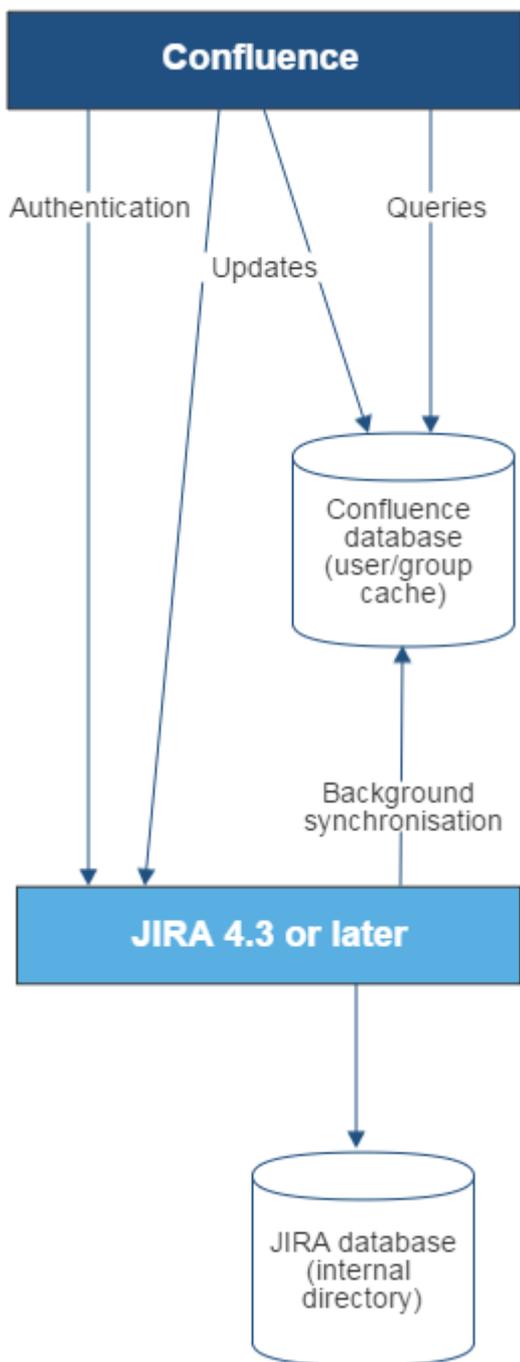


Diagram above: Confluence connecting to JIRA for user management.

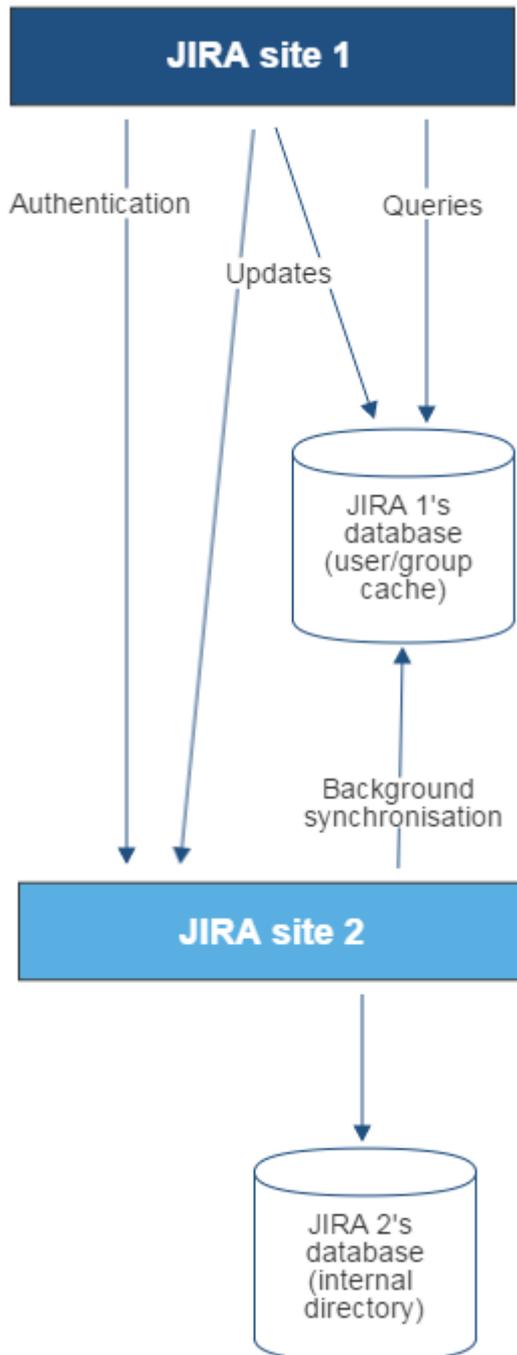


Diagram above: One JIRA site connecting to another for user management. JIRA site 2 does the user management, storing the user data in its internal directory.

Related topics

### Configuring user directories

- [Configuring the internal directory](#)
- [Connecting to an LDAP directory](#)
- [Connecting to an internal directory with LDAP authentication](#)
- [Connecting to Crowd or another Jira application for user management](#)
- [Managing multiple directories](#)
- [Migrating users between user directories](#)
- [Synchronizing data from external directories](#)

# Diagrams of possible configurations for user management

The aim of these diagrams is to help people understand each directory type at a glance. We have kept the diagrams simple and conceptual, with just enough information to be correct.

Some things that we **do not** attempt to show:

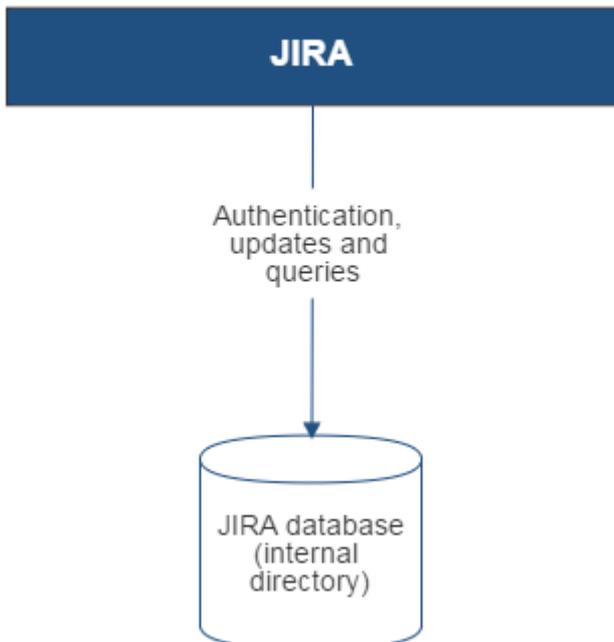
- In most cases, we do not attempt to show that you can have multiple directory types mapped to Jira at the same time. We illustrate that fact in just the first two LDAP diagrams.
- We have not included a diagram for Confluence's legacy connection to Jira database.
- We do not attempt to show all of the possible configurations and layered connections that are available now that you can use Jira as a directory manager.

**i** Managing 500+ users across Atlassian products? Find out how easy, scalable and effective it can be with Crowd! See [centralized user management](#).

**On this page:**

- [Jira internal directory](#)
- [Jira with read/write connection to LDAP](#)
- [Jira with read-only connection to LDAP, with local groups](#)
- [Jira internal directory with LDAP authentication](#)
- [Jira with LDAP authentication, copy users on first login](#)
- [One Jira instance connecting to another](#)
- [Confluence and Jira connecting to Crowd](#)
- [A number of applications connecting to Jira](#)

Jira internal directory



*Diagram above: JIRA using its internal directory for user management.*

Jira with read/write connection to LDAP

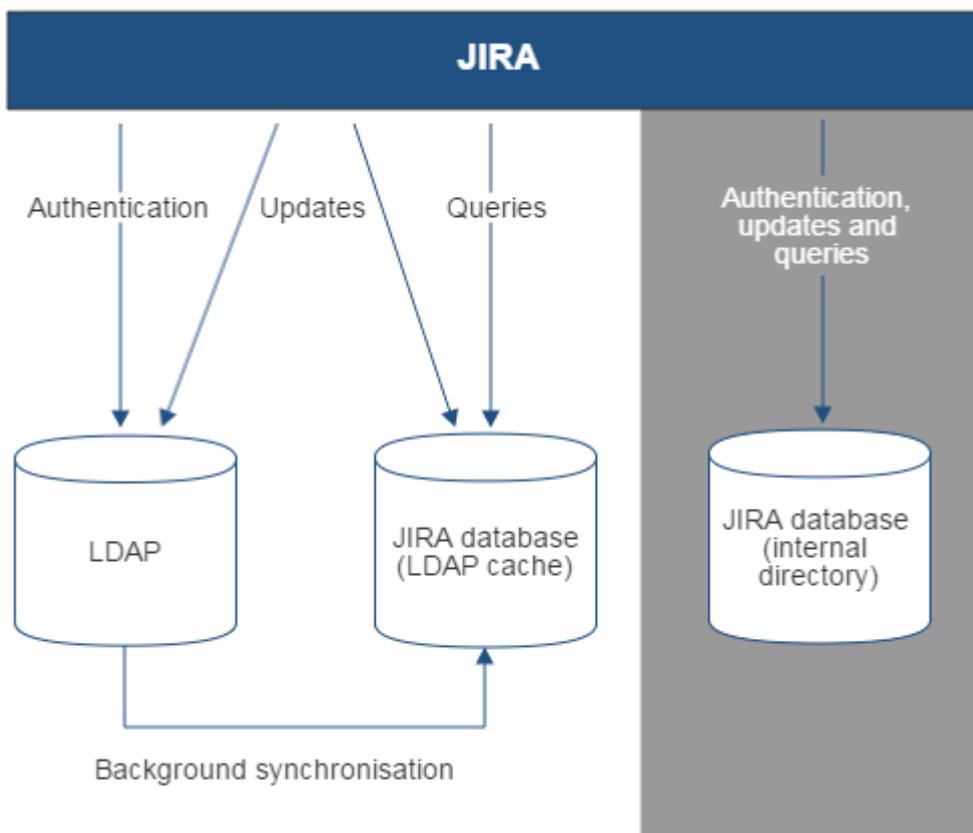


Diagram above: JIRA connecting to an LDAP directory.

Jira with read-only connection to LDAP, with local groups

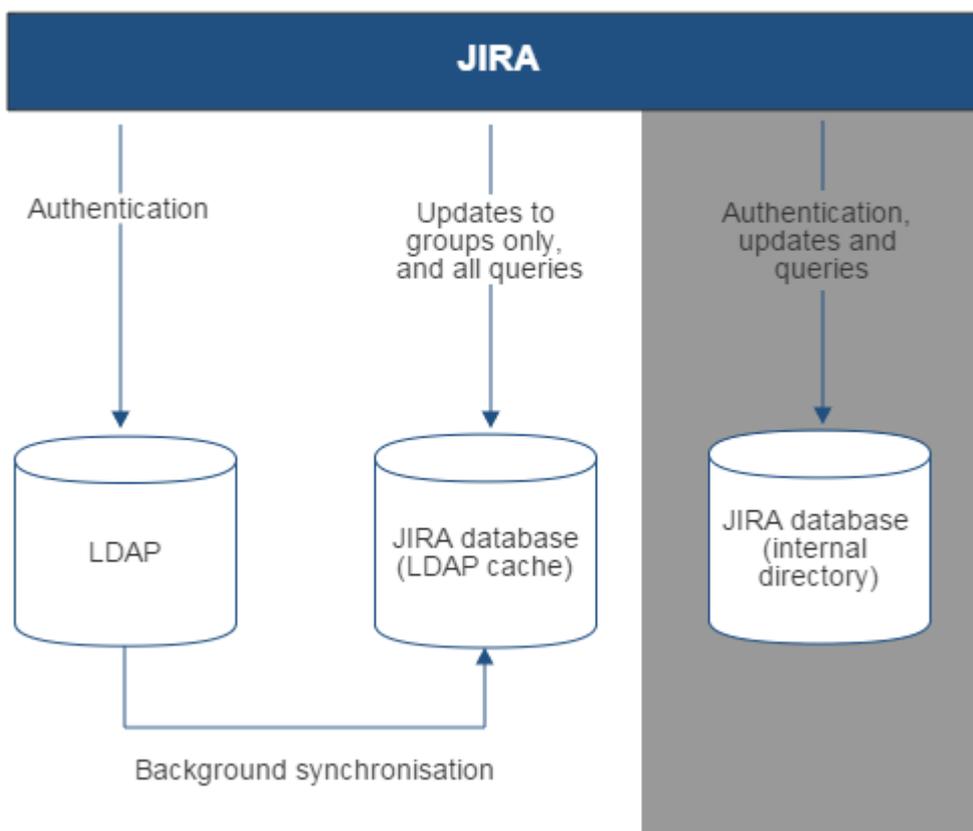


Diagram above: JIRA connecting to an LDAP directory with permissions set to read only and local groups.

Jira internal directory with LDAP authentication

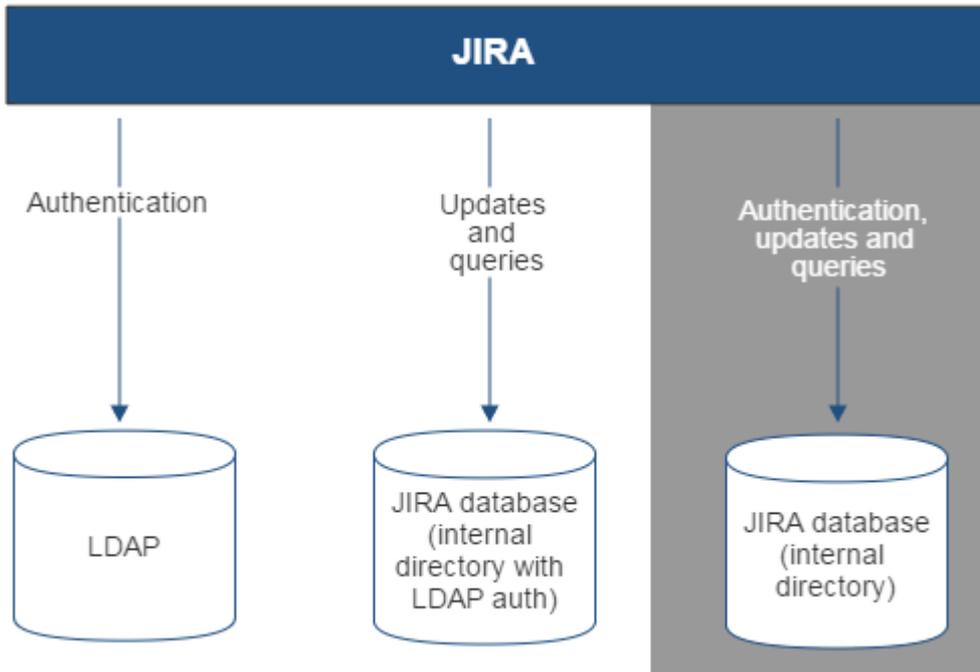


Diagram above: JIRA connecting to an LDAP directory for authentication only.

Jira with LDAP authentication, copy users on first login

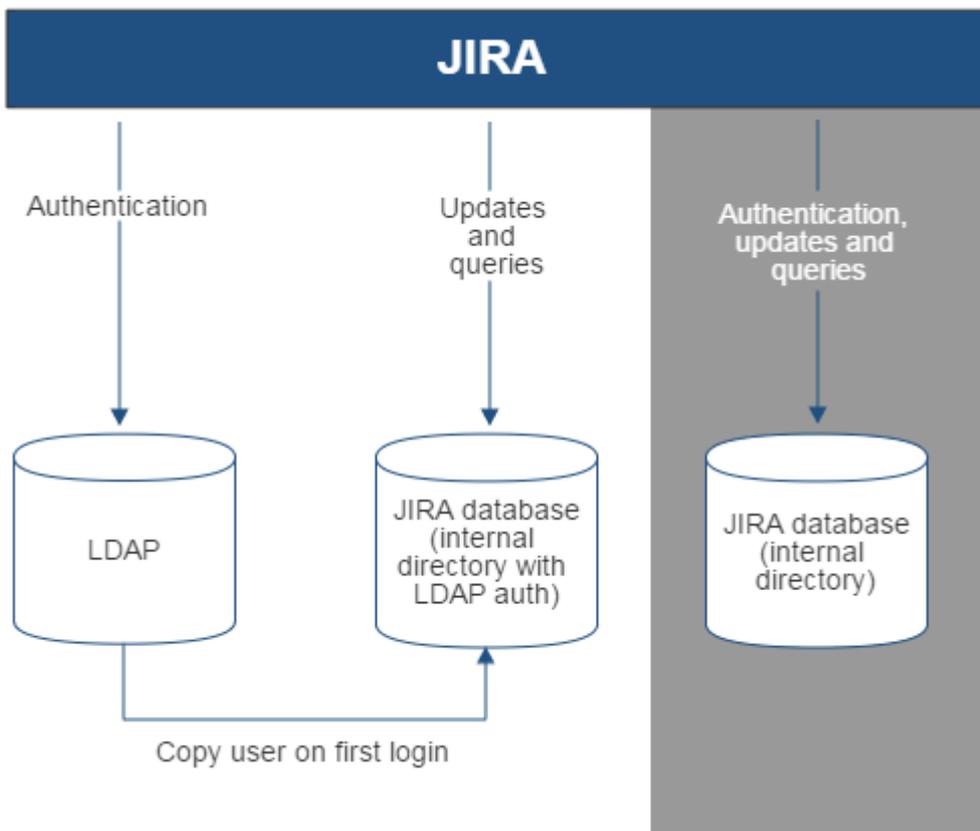


Diagram above: JIRA connecting to an LDAP directory for authentication only, with each user copied to the internal directory when they first log in to JIRA.

One Jira instance connecting to another

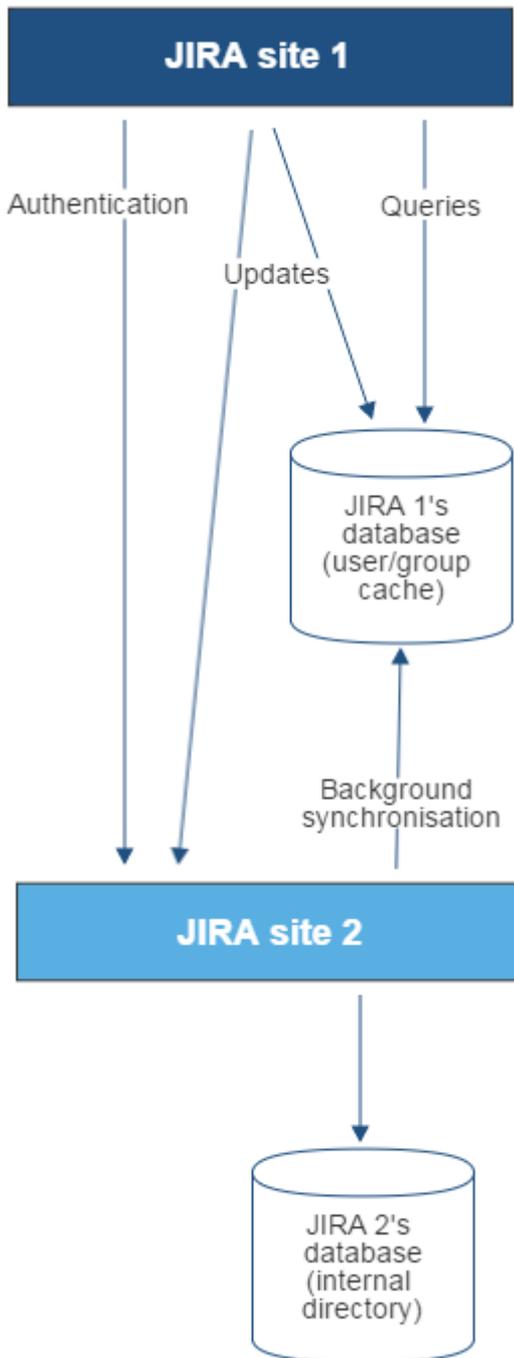


Diagram above: One JIRA site connecting to another for user management. JIRA site 2 does the user management, storing the user data in its internal directory.

Confluence and Jira connecting to Crowd

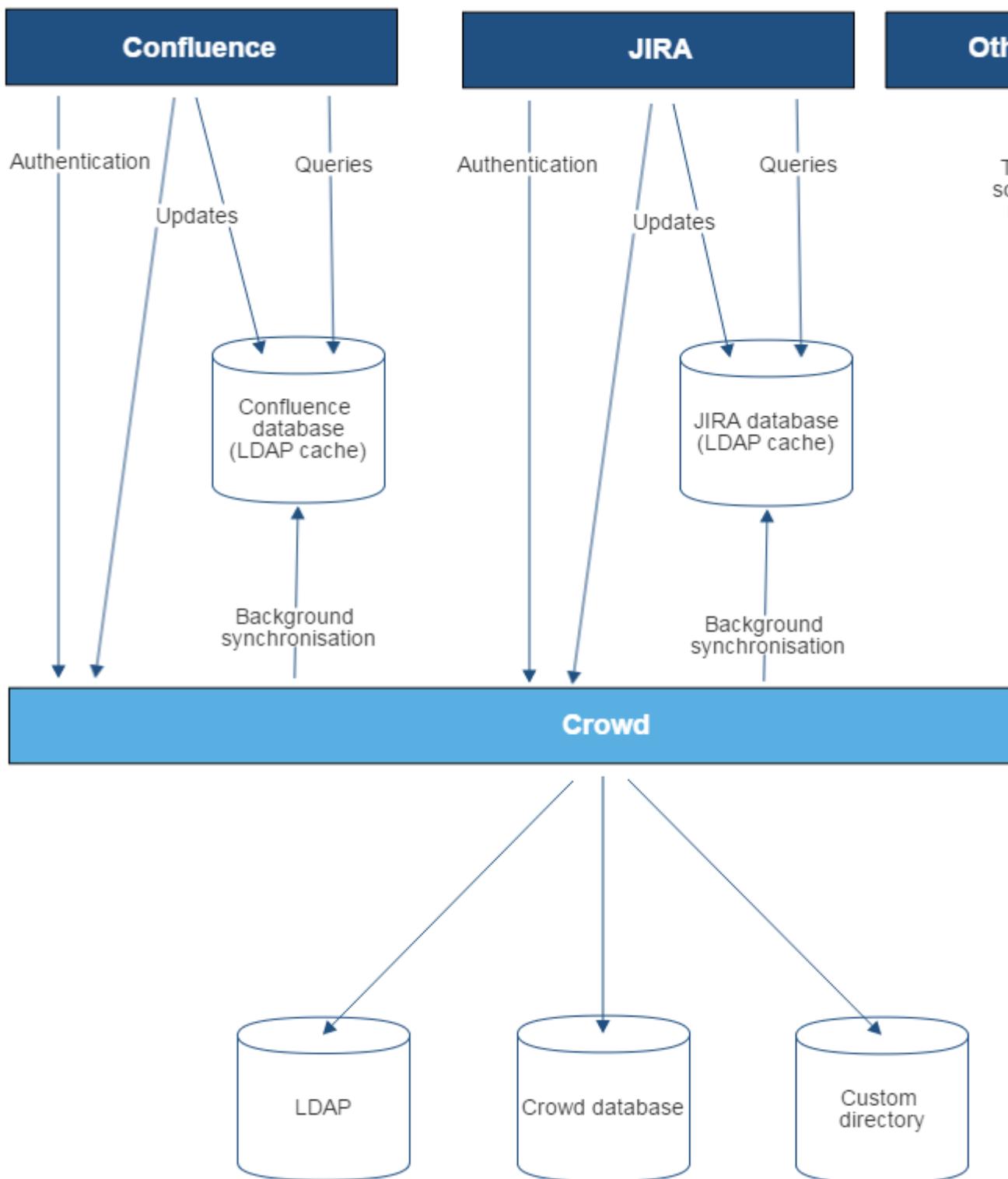


Diagram above: Confluence, JIRA and other applications connecting to Crowd for user management.

A number of applications connecting to Jira

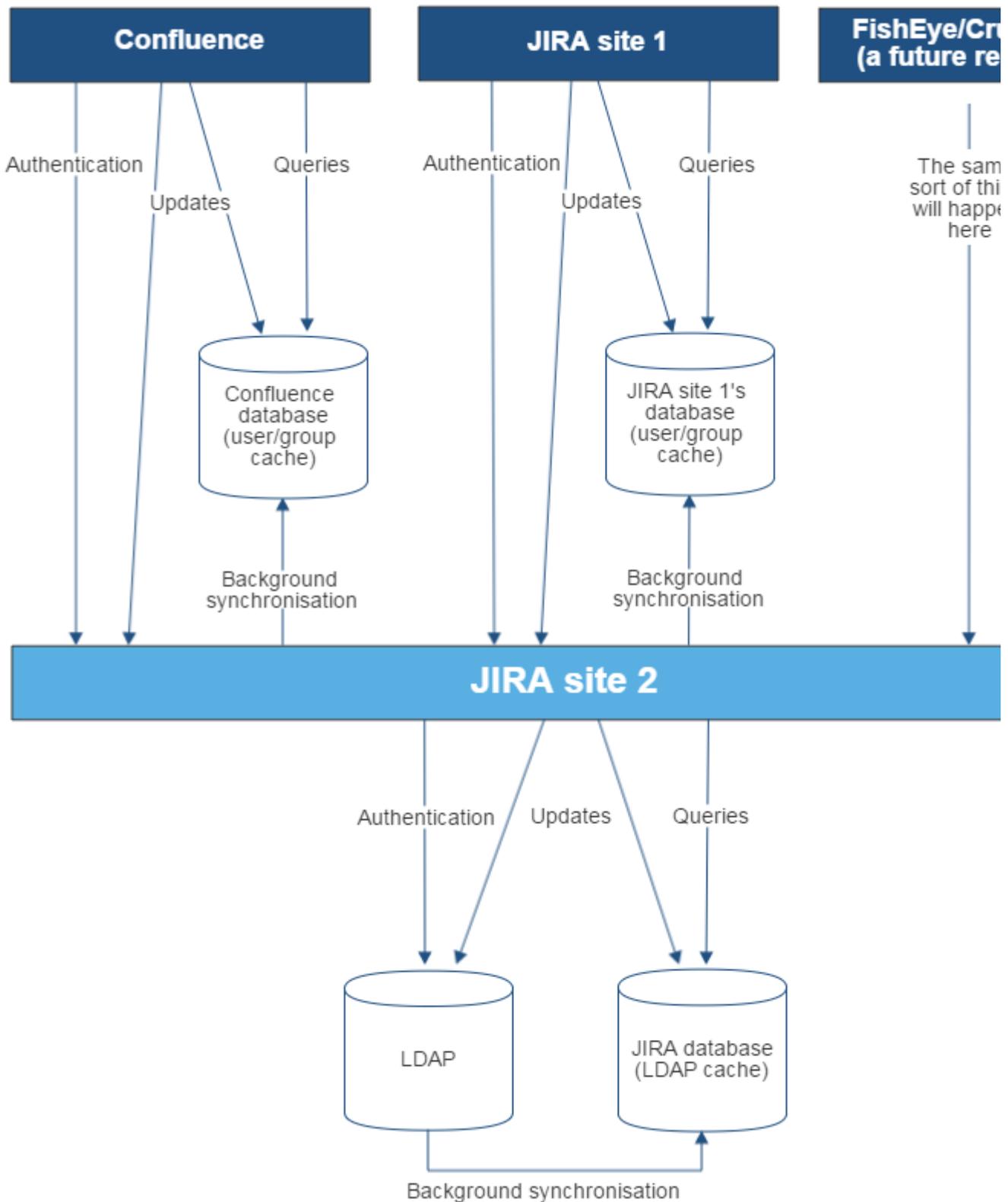


Diagram above: A number of applications connecting to JIRA (site 2) for user management, with JIRA in turn connecting to an LDAP server.

Related topics

#### Configuring user directories

- [Configuring the internal directory](#)
- [Connecting to an LDAP directory](#)

- [Connecting to an internal directory with LDAP authentication](#)
- [Connecting to Crowd or another Jira application for user management](#)
- [Managing multiple directories](#)
- [Migrating users between user directories](#)
- [Synchronizing data from external directories](#)

# Managing nested groups

Some directory servers allow you to define a group as a member of another group. Groups in such a structure are called *nested groups*. Nested groups simplify permissions by allowing sub-groups to inherit permissions from a parent group.

This page describes how Jira handles nested groups that exist in one or more of your directory servers. Note that if you're using nested groups, you can't use an [LDAP directory for delegated authentication](#).

## Enabling Nested Groups

You can enable or disable support for nested groups on each directory individually. Select **User Directories** from the Jira administration menu, **edit** the directory and select **Enable Nested Groups**. See [Configuring user directories](#).

Notes:

- Make sure that your directory server supports nested groups before you enable nested groups for a specific directory type in Jira.
- You can nest internal groups in internal groups, or external groups in external groups. You can't nest an internal group in an external group or vice versa.
- Please read the rest of this page to understand what effect nested groups will have on authentication (login) and permissions in Jira, and what happens when you update users and groups in Jira.



Managing 500+ users across Atlassian products? Find out how easy, scalable and effective it can be with Crowd! See [centralized user management](#).

### On this page:

- [Enabling Nested Groups](#)
- [Effect of Nested Groups](#)
  - [Login](#)
  - [Permissions](#)
  - [Viewing lists of group members](#)
  - [Adding and updating group membership](#)
- [Examples](#)
  - [Example 1: User is member of sub-group](#)
  - [Example 2: Sub-groups as members of the jira-developers group](#)
- [Notes](#)

## Effect of Nested Groups

This section explains how nested groups affect logging in, permissions, and viewing and updating users and groups.

### Login

When a user logs in, they can access the application if they belong to an authorized group or any of its sub-groups.

### Permissions

The user can access a function if they belong to a group that has the necessary permissions, or if they belong to any of its sub-groups.

## Viewing lists of group members

If you ask to view the members of a group, you will see all users who are members of the group and all users belonging its sub-groups, consolidated into one list. We call this a *flattened* list.

You can't view or edit the nested groups themselves, or see that one group is a member of another group.

### Adding and updating group membership

If you add a user to a group, the user is added to the named group and not to any other groups.

If you try to remove a user from a [flattened](#) list, the following will happen:

- If the user is a member of the top group in the hierarchy of groups in the flattened list, the user is removed from the top group.
- Otherwise, you see an error message stating that the user is not a direct member of the group.

### Examples

#### Example 1: User is member of sub-group

Imagine the following two groups exist in your directory server:

- staff
- marketing

Memberships:

- The **marketing** group is a member of the **staff** group.
- User **jsmith** is a member of **marketing**.

You will see that **jsmith** is a member of both **marketing** and **staff**. You will not see that the two groups are nested. If you assign permissions to the **staff** group, then **jsmith** will get those permissions.

#### Example 2: Sub-groups as members of the jira-developers group

In an LDAP directory server, we have the groups **engineering-group** and **techwriters-group**. We want to grant both groups developer-level access to the JIRA. We will have a group called **jira-developers** that has developer-level access.

- Add a group called **jira-developers**.
- Add the **engineering-group** as a sub-group of **jira-developers**.
- Add the **techwriters-group** as a sub-group of **jira-developers**.

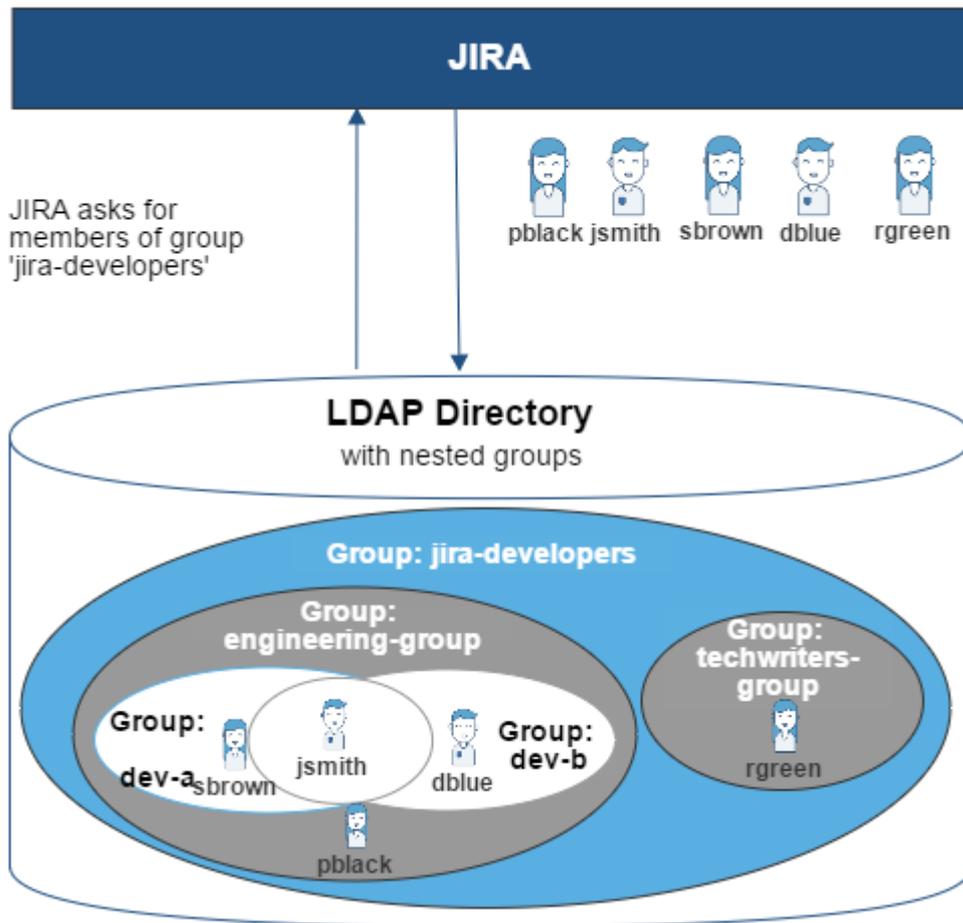
Group memberships are now:

- **jira-developers** sub-groups: **engineering-group**, **techwriters-group**
- **engineering-group** sub-groups: **dev-a**, **dev-b**; users: **pblack**
- **dev-a** users: **jsmith**, **sbrown**
- **dev-b** users: **jsmith**, **dblue**
- **techwriters-group** users: **rgreen**

When the JIRA application requests a list of users in the **jira-developers** group, it receives the following list:

- **pblack**
- **jsmith**
- **sbrown**
- **dblue**
- **rgreen**

*Diagram: Sub-groups as members of the **jira-developers** group*



## Notes

- **Possible impact on performance.** Enabling nested groups may result in slower user searches.
- **Definition of nested groups in LDAP.** In an LDAP directory, a nested group is a child group entry whose DN (Distinguished Name) is referenced by an attribute contained within a parent group entry. For example, a parent group **Group One** might have an `objectClass=group` attribute and one or more `member=DN` attributes, where the DN can be that of a user or that of a group elsewhere in the LDAP tree:

```
member=CN=John Smith,OU=Users,OU=OrgUnitA,DC=sub,DC=domain
member=CN=Group Two,OU=OrgUnitBGroups,OU=OrgUnitB,DC=sub,DC=domain
```

## Related topics

### Configuring user directories

- [Configuring the internal directory](#)
- [Connecting to an LDAP directory](#)
- [Connecting to an internal directory with LDAP authentication](#)
- [Connecting to Crowd or another Jira application for user management](#)
- [Managing multiple directories](#)
- [Migrating users between user directories](#)
- [Synchronizing data from external directories](#)

# User management limitations and recommendations

This page describes the optimal configurations and limitations that apply to user management in Jira.

## On this page:

- [Recommendations for connecting to LDAP](#)
- [Recommendations for connecting to another Jira server](#)

## General recommendations

- **Avoid duplicate usernames across directories.** If you are connecting to more than one user directory, we recommend that you ensure the usernames are unique to one directory. For example, we do not recommend that you have a user `jsmith` in both 'Directory1' and 'Directory2'. The reason is the potential for confusion, especially if you swap the order of the directories. Changing the directory order can change the user that a given username refers to.
- **Be careful when deleting users in remote directories.** If you are connecting to an LDAP directory, a Crowd directory or a remote Jira directory, please take care when deleting users from the remote directory. If you delete a user that is associated with data in Jira, this will cause problems in Jira. We recommend that you perform all user management in Jira, because the Jira UI will prevent the deletion of a user if there are issues assigned to the user, reported by the user or the user is a project lead.

 Managing 500+ users across Atlassian products? Find out how easy, scalable and effective it can be with Crowd! See [centralized user management](#).

## Recommendations for connecting to LDAP

Please consider the following limitations and recommendations when connecting to an LDAP user directory.

### Optimal Number of Users and Groups in your LDAP Directory

The connection to your LDAP directory provides powerful and flexible support for connecting to, configuring and managing LDAP directory servers. To achieve optimal performance, a background synchronization task loads the required users and groups from the LDAP server into the application's database, and periodically fetches updates from the LDAP server to keep the data in step. The amount of time needed to copy the users and groups rises with the number of users, groups, and group memberships. For that reason, we recommended a maximum number of users and groups as described below.

This recommendation affects connections to LDAP directories:

- Microsoft Active Directory
- All other LDAP directory servers

The following LDAP configurations are **not** affected:

- Internal directories with LDAP authentication
- LDAP directories configured for 'Authentication Only, Copy User On First Login'

Please choose one of the following solutions, depending on the number of users, groups and memberships in your LDAP directory.

Your environment	Recommendation
Up to 10 000 (ten thousand) users, 1000 (one thousand) groups, and 20 (twenty) groups per user	Choose the 'LDAP' or 'Microsoft Active Directory' directory type. You can make use of the full synchronization option. Your application's database will contain all the users and groups that are in your LDAP server.
More than the above	Use LDAP filters to reduce the number of users and groups visible to the synchronization task.

#### Our Test Results

We performed internal testing of synchronization with an AD server on our local network consisting of 10 000 users, 1000 groups and 200 000 memberships.

We found that the initial synchronization took about 5 minutes. Subsequent synchronizations with 100 modifications on the AD server took a couple of seconds to complete.

Please keep in mind that a number of factors come into play when trying to tune the performance of the synchronization process, including:

- **Size of userbase.** Use LDAP filters to keep this to the minimum that suits your requirements.
- **Type of LDAP server.** We currently support change detection in AD, so subsequent synchronizations are much faster for AD than for other LDAP servers.
- **Network topology.** The further away your LDAP server is from your application server, the more latent LDAP queries will be.
- **Database performance.** As the synchronization process caches data in the database, the performance of your database will affect the performance of the synchronization.
- **JVM heap size.** If your heap size is too small for your userbase, you may experience heavy garbage collection during the synchronization process which could in turn slow down the synchronization.

#### Redundant LDAP is Not Supported

The LDAP connections do not support the configuration of two or more LDAP servers for redundancy (automated failover if one of the servers goes down).

#### Specific Notes for Connecting to Active Directory

When the application synchronizes with Active Directory (AD), the synchronization task requests only the changes from the LDAP server rather than the entire user base. This optimizes the synchronization process and gives much faster performance on the second and subsequent requests.

On the other hand, this synchronization method results in a few limitations:

1. **Externally moving objects out of scope or renaming objects causes problems in AD.** If you move objects out of scope in AD, this will result in an inconsistent cache. We recommend that you do not use the external LDAP directory interface to move objects out of the scope of the sub-tree, as defined on the application's directory configuration screen. If you do need to make structural changes to your LDAP directory, manually synchronize the directory cache after you have made the changes to ensure cache consistency.
2. **Synchronizing between AD servers is not supported.** Microsoft Active Directory does not replicate the uSNChanged attribute across instances. For that reason, we do not support connecting to different AD servers for synchronization. (You can of course define multiple different directories, each pointing to its own respective AD server.)
3. **Synchronizing with AD servers behind a load balancer is not supported.** As with synchronizing between two different AD servers, Microsoft Active Directory does not replicate the uSNChanged attribute across instances. For that reason, we do not support connecting to different AD servers even when they are load balanced. You will need to select one server (preferably one that is local) to synchronize with instead of using the load balancer.
4. **You must restart the application after restoring AD from backup.** On restoring from backup of an AD server, the uSNChanged timestamps are reverted to the backup time. To avoid the resulting confusion, you will need to flush the directory cache after a Active Directory restore operation.

5. **Obtaining AD object deletions requires administrator access.** Active Directory stores deleted objects in a special container called cn=Deleted Objects. By default, to access this container you need to connect as an administrator and so, for the synchronization task to be aware of deletions, you must use administrator credentials. Alternatively, it is possible to change the permissions on the cn=Deleted Objects container. If you wish to do so, please see [this Microsoft KB article](#).
6. **The User DN used to connect to AD must be able to see the uSNChanged attribute.** The synchronization task relies on the uSNChanged attribute to detect changes, and so must be in the appropriate AD security groups to see this attribute for all LDAP objects in the subtree.

## Recommendations for connecting to another Jira server

Please consider the following limitations and recommendations when connecting to a JIRA server for user management.

### Single Sign-On Across Multiple Applications is Not Supported

When you connect to a JIRA application for user management, you will not have single sign-on across the applications connected in this way. JIRA, when acting as a directory manager, does not support SSO.

### Custom Application Connectors are Not Supported

JIRA applications, Confluence, FishEye, Crucible and Bamboo can connect to a JIRA server for user management. Custom application connectors will need to use the new REST API.

### Custom Directories are Not Supported

Earlier versions of JIRA supported OSUser Providers. It was therefore possible write a special provider to obtain user information from any external user directory. This is no longer the case.

### Load on your JIRA instance

If your JIRA instance is already under high load, then using it as a User Server will increase that load.

### JIRA Cloud applications not supported

You cannot use JIRA Cloud applications to manage standalone users. Cloud users and users within your self-hosted Atlassian applications need to be managed separately.

## Recommendations

Your environment	Recommendation
<p>If <b>all</b> the following are true:</p> <ul style="list-style-type: none"> <li>• Your JIRA application is not under high load.</li> <li>• You want to share user and group management across just a few applications, such as one JIRA Software server and one Confluence server, or two JIRA servers.</li> <li>• You do not need single sign-on (SSO) between your JIRA application and Confluence, or between two JIRA servers.</li> <li>• You do not have custom application connectors. Or, if you do have them, you are happy to convert them to use the new REST API.</li> <li>• You are happy to shut down all your servers when you need to upgrade your JIRA application.</li> </ul>	<p>Your environment meets the optimal requirements for using a JIRA application for user management.</p>
<p>If <b>one or more</b> of the following are true:</p>	

<ul style="list-style-type: none"> <li>• If your JIRA application is already under high load.</li> <li>• You want to share user and group management across more than 5 applications.</li> <li>• You need single sign-on (SSO) across multiple applications.</li> <li>• You have custom applications integrated via the Crowd SOAP API, and you cannot convert them to use the new REST API.</li> <li>• You are not happy to shut down all your servers when you need to upgrade JIRA.</li> </ul>	<p>We recommend that you install <a href="#">Atlassian Crowd</a> for user management and SSO.</p>
<p>If you are considering creating a custom directory connector to define your own storage for users and groups...</p>	<p>Please see if one of the following solutions will work for you:</p> <ul style="list-style-type: none"> <li>• If you have written a custom provider to support a specific LDAP schema, please check the supported LDAP schemas to see if you can use one of them instead.</li> <li>• If you have written a custom provider to support nested groups, please consider enabling nested groups in the supported directory connectors instead.</li> <li>• If you have written a custom provider to connect to your own database, please consider loading the data into the application's database instead.</li> <li>• If you need to keep the custom directory connection, please consider whether <a href="#">Atlassian Crowd</a> meets your requirements. See the documentation on <a href="#">Creating a Custom Directory Connector</a>.</li> </ul>

#### Related topics

[Connecting to an LDAP directory](#)

[Connecting to Crowd or another Jira server for user management](#)

[Configuring user directories](#)

# Configuring user directories

A user directory is a place where you store information about users and groups. User information includes the person's full name, username, password, email address and other personal information. Group information includes the name of the group, the users that belong to the group, and possibly groups that belong to other groups.

The **internal** directory stores user and group information in the Jira database. You can also connect to **external** user directories, and to Atlassian **Crowd** and **Jira** as directory managers.

## On this page:

- [Configuring user directories in Jira](#)
- [Connecting to a directory](#)
- [Updating directories](#)

 Managing 500+ users across Atlassian products?  
Find out how easy, scalable and effective it can be with Crowd!  
See [centralized user management](#).

## Configuring user directories in Jira

1. Log in as a user with the '**Jira System Administrators**' [global permission](#).
2. Choose **Administration**  **> User Management > User Directories**.

## Connecting to a directory

You can add the following types of directory servers and directory managers:

- Jira's internal directory. See [Configuring the internal directory](#).
- Microsoft Active Directory. See [Connecting to an LDAP directory](#).
- Various other LDAP directory servers. See [Connecting to an LDAP Directory](#).
- An LDAP directory for delegated authentication. See [Connecting to an Internal Directory with LDAP Authentication](#).
- Atlassian Crowd. See [Connecting to Crowd or another Jira server for user management](#).
- Another Jira server. See [Connecting to Crowd or another Jira server for user management](#).

You can add as many external user directories as you need. Note that you can define the **order** of the directories. This determines which directory Jira will search first, when looking for user and group information. See [Managing multiple directories](#).

## Updating directories

### Limitations when Editing Directories

You cannot edit, disable or remove the directory your user belongs to. This precaution is designed to prevent administrators from locking themselves out of the application by changing the directory configuration in a way that prevents them logging in or removes their administration permissions.

This limitation applies to all directory types. For example:

- You cannot disable the internal directory if your user is an internal user.
- You cannot disable or remove an LDAP or a Crowd directory if your user comes from that directory.

In some situations, reordering the directories will change the directory that the current user comes from, if a user with the same username happens to exist in both. This behavior can be used in some cases to create a copy of the existing configuration, move it to the top, then remove the old one. Note, however, that duplicate usernames are not a supported configuration.

You cannot remove the internal directory. This precaution aligns with the recommendation below that you always keep an administrator account active in the internal directory.

### Recommendations

The recommended way to edit directory configurations is to log in as an internal user when making changes to external directory configuration.

 We recommend that you keep either an administrator or system administrator user active in your internal directory for troubleshooting problems with your user directories.

### Enabling, disabling, and removing directories

You can enable or disable a directory at any time. If you disable a directory, your configuration details will remain but the application will not recognize the users and groups in that directory.

You have to disable a directory before you can remove it. Removing a directory will remove the details from the database.

[Screenshot: Configuring user directories](#)

## User Directories

The table below shows the user directories currently configured for JIRA.

The order of the directories is the order in which they will be searched for users and groups. Changes to users and groups will be made in the first directory where JIRA has permission to make changes. It is recommended that users only exist in a single directory.

Directory Name	Type	Order	Operations
JIRA Internal Directory	Internal	↑ ↓	
LDAP server	OpenLDAP (Read-Write)	↑ ↓	<a href="#">Disable</a> <a href="#">Edit</a> <a href="#">Synchronise</a> Last synchronised at 17/01/11 10:31 AM (took 72s).

[Add Directory](#)

 In situations where users are unable to change their passwords, check that a Delegated Authentication Directory is not the highest in the order of User Directories. As a workaround, you can change the order of User Directories, or alternatively use a connection to a LDAP directory instead.

### Related topics

- [Configuring the internal directory](#)
- [Connecting to an LDAP directory](#)
- [Connecting to an internal directory with LDAP authentication](#)
- [Connecting to Crowd or another Jira application for user management](#)
- [Managing multiple directories](#)
- [Migrating users between user directories](#)

- [Synchronizing data from external directories](#)
- [User management](#)

# Configuring the internal directory

The internal directory stores user and group information in the Jira database.

The internal directory is enabled by default at installation. When you create the first administrator during the setup procedure, that administrator's username and other details are stored in the internal directory.

If needed, you can configure one or more additional user directories. This is useful if you want to grant access to users and groups that are stored in a corporate directory or other directory server.

## On this page:

- [Settings](#)
- [Diagram of possible configuration](#)

**i** Managing 500+ users across Atlassian products? Find out how easy, scalable and effective it can be with Crowd! See [centralized user management](#).

## Settings

Setting	Description
Enable Nested Groups	Enable or disable support for nested groups. When nested groups are enabled, you can define a group as a member of another group. If you are using groups to manage permissions, you can create nested groups to allow inheritance of permissions from one group to its sub-groups.

## Diagram of possible configuration

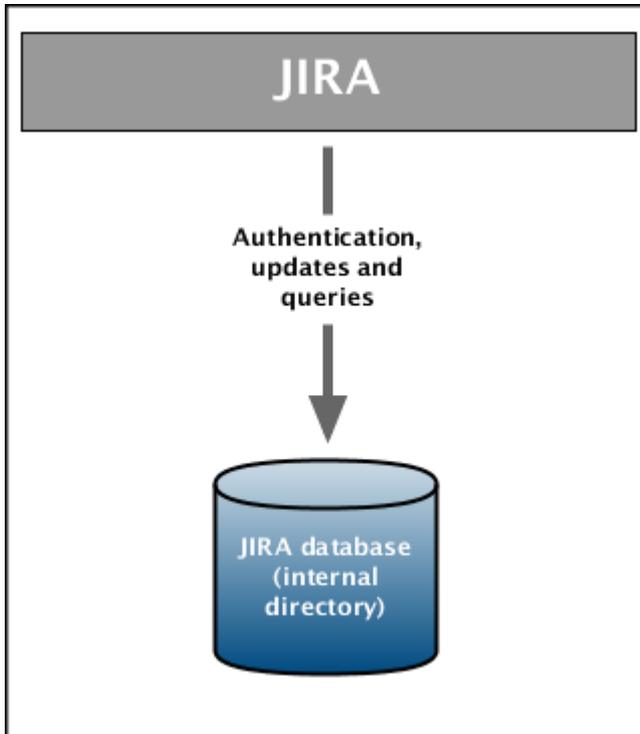


Diagram above: JIRA using its internal directory for user management.

Related topics

### [Configuring user directories](#)

- [Configuring the internal directory](#)

- [Connecting to an LDAP directory](#)
- [Connecting to an internal directory with LDAP authentication](#)
- [Connecting to Crowd or another Jira application for user management](#)
- [Managing multiple directories](#)
- [Migrating users between user directories](#)
- [Synchronizing data from external directories](#)

# Connecting to an LDAP directory

You can connect your Jira application to an LDAP directory for authentication, user and group management. You will need to log in as a user with the Jira System Administrators' [global permission](#) to access the Settings menu below.

 Managing 500+ users across Atlassian products? Find out how easy, scalable and effective it can be with Crowd! See [centralized user management](#).

An LDAP directory is a collection of data about users and groups. LDAP (Lightweight Directory Access Protocol) is an Internet protocol that web applications can use to look up information about those users and groups from the LDAP server.

We provide built-in connectors for the most popular LDAP directory servers:

- Microsoft Active Directory
- Apache Directory Server (ApacheDS)
- Apple Open Directory
- Fedora Directory Server
- Novell eDirectory
- OpenDS
- OpenLDAP
- OpenLDAP Using Posix Schema
- Posix Schema for LDAP
- Sun Directory Server Enterprise Edition (DSEE)
- A generic LDAP directory server

**When to use this option:** Connecting to an LDAP directory server is useful if your users and groups are stored in a corporate directory. When configuring the directory, you can choose to make it read only, read only with local groups, or read/write. If you choose read/write, any changes made to user and group information in the application will also update the LDAP directory.

 Learn more about [synchronizing data from external directories](#).

## On this page:

- [Connecting to an LDAP Directory in Jira](#)
- [Server settings](#)
- [Schema settings](#)
- [Permission settings](#)
  - [Adding user to groups automatically](#)
- [Advanced settings](#)
- [User schema settings](#)
- [Group schema settings](#)
- [Membership schema settings](#)
- [Diagrams of some possible configurations](#)

## Connecting to an LDAP Directory in Jira

1. Choose **Administration**  > **User Management**.
2. Choose **User Directories**.
3. **Add** a directory and select one of these types:
  - **'Microsoft Active Directory'** This option provides a quick way to select AD, because it is the most popular LDAP directory type.
  - **'LDAP'** You will be able to choose a specific LDAP directory type on the next screen.
4. Enter the values for the settings, as described below.
5. Save the directory settings.
6. Define the **directory order** by clicking the blue up- and down-arrows next to each directory on the **'User Directories'** screen. Here is a summary of how the directory order affects the processing:
  - The order of the directories is the order in which they will be searched for users and groups.
  - Changes to users and groups will be made only in the first directory where the application has permission to make changes.

For details, see [Managing multiple directories](#).

## Notes:

- For this configuration, every time user logs in (i.e. first and subsequent times), the user's data in Jira will be updated from the user's data in LDAP. This includes username, display name, email and group memberships. However for group memberships, only the following applies:
  - direct groups only (i.e. not nested groups) are synchronized from LDAP.
  - only groups that are already present in Jira are synchronized, i.e. groups are not added /removed, and group hierarchies are not synchronized.

 Learn more about [synchronizing data from external directories](#).

## Server settings

Setting	Description
Name	Enter a meaningful name to help you identify the LDAP directory server. Examples: <ul style="list-style-type: none"> <li>• Example Company Staff Directory</li> <li>• Example Company Corporate LDAP</li> </ul>
Directory Type	Select the type of LDAP directory that you will connect to. If you are adding a new LDAP connection, the value you select here will determine the default values for many of the options on the rest of screen. Examples: <ul style="list-style-type: none"> <li>• Microsoft Active Directory</li> <li>• OpenDS</li> <li>• And more.</li> </ul>
Hostname	The host name of your directory server. Examples: <ul style="list-style-type: none"> <li>• ad.example.com</li> <li>• ldap.example.com</li> <li>• opens.example.com</li> </ul>
Port	The port on which your directory server is listening. Examples: <ul style="list-style-type: none"> <li>• 389</li> <li>• 10389</li> <li>• 636 (for example, for SSL)</li> </ul>
Use SSL	Check this if the connection to the directory server is an SSL (Secure Sockets Layer) connection. Note that you will need to configure an SSL certificate in order to use this setting.

Username	<p>The distinguished name of the user that the application will use when connecting to the directory server. Examples:</p> <ul style="list-style-type: none"> <li>• <code>cn=administrator,cn=users,dc=ad,dc=example,dc=com</code></li> <li>• <code>cn=user,dc=domain,dc=name</code></li> <li>• <code>user@domain.name</code></li> </ul> <div style="border: 1px solid red; padding: 10px; margin-top: 10px;"> <p> By default, all users can read the <code>uSNChanged</code> attribute; however, only administrators or users with relevant permissions can access the Deleted Objects container. The specific privileges required by the user to connect to LDAP are "Bind" and "Read" (user info, group info, group membership, update sequence number, deleted objects), which the user can obtain by being a member of the Active Directory's built-in administrators group.</p> <p>Note that the incremental sync will fail silently if the Active Directory is accessed by a user without these privileges. This has been reported as <a href="#">CWD-3093</a>.</p> </div>
Password	<p>The password of the user specified above.</p> <p><b>Note:</b> Connecting to an LDAP server requires that this application log in to the server with the username and password configured here. As a result, this password cannot be one-way hashed - it must be recoverable in the context of this application. The password is currently stored in the database in plain text without obfuscation. To guarantee its security, you need to ensure that other processes do not have OS-level read permissions for this application's database or configuration files.</p>

## Schema settings

Setting	Description
Base DN	<p>The root distinguished name (DN) to use when running queries against the directory server. Examples:</p> <ul style="list-style-type: none"> <li>• <code>o=example,c=com</code></li> <li>• <code>cn=users,dc=ad,dc=example,dc=com</code></li> <li>• For Microsoft Active Directory, specify the base DN in the following format: <code>dc=domain1,dc=local</code>. You will need to replace the <code>domain1</code> and <code>local</code> for your specific configuration. Microsoft Server provides a tool called <code>ldp.exe</code> which is useful for finding out and configuring the the LDAP structure of your server.</li> </ul>
Additional User DN	<p>This value is used in addition to the base DN when searching and loading users. If no value is supplied, the subtree search will start from the base DN. Example:</p> <ul style="list-style-type: none"> <li>• <code>ou=Users</code></li> </ul>
Additional Group DN	<p>This value is used in addition to the base DN when searching and loading groups. If no value is supplied, the subtree search will start from the base DN. Example:</p> <ul style="list-style-type: none"> <li>• <code>ou=Groups</code></li> </ul>



If no value is supplied for **Additional User DN** or **Additional Group DN** this will cause the subtree search to start from the base DN and, in case of huge directory structure, could cause performance issues for login and operations that rely on login to be performed.

## Permission settings

**Note:** You can only assign LDAP users to local groups when 'External User Management' is not selected.

Setting	Description
Read Only	LDAP users, groups and memberships are retrieved from your directory server and can only be modified via your directory server. You cannot modify LDAP users, groups or memberships via the application administration screens.
Read Only, with Local Groups	<p>LDAP users, groups and memberships are retrieved from your directory server and can only be modified via your directory server. You cannot modify LDAP users, groups or memberships via the application administration screens. However, you can add groups to the internal directory and add LDAP users to those groups.</p> <p><b>Note for Confluence users:</b> Users from LDAP are added to groups maintained in Confluence's internal directory the first time they log in. This is only done once per user. There is a known issue with Read Only, with Local Groups in Confluence that may apply to you. See <a href="#">CONFSERVER-28624</a> - User Loses all Local Group Memberships If LDAP Sync is Unable to find the User, but the User appears again in subsequent syncs <span>CLOSED</span></p>
Read /Write	LDAP users, groups and memberships are retrieved from your directory server. When you modify a user, group or membership via the application administration screens, the changes will be applied directly to your LDAP directory server. Please ensure that the LDAP user specified for the application has modification permissions on your LDAP directory server.

### Adding user to groups automatically

Setting	Description
Default Group Memberships	<p><i>Option available in Confluence 3.5 and later, and JIRA 4.3.3 and later.</i> This field appears if you select the 'Read Only, with Local Groups' permission. If you would like users to be automatically added to a group or groups, enter the group name(s) here. To specify more than one group, separate the group names with commas.</p> <p><i>In Confluence 3.5 to Confluence 3.5.1:</i> Each time a user logs in, their group memberships will be checked. If the user does not belong to the specified group(s), their username will be added to the group(s). If a group does not yet exist, it will be added locally.</p> <p><i>In Confluence 3.5.2 and later, and JIRA 4.3.3 and later:</i> The first time a user logs in, their group memberships will be checked. If the user does not belong to the specified group(s), their username will be added to the group(s). If a group does not yet exist, it will be added locally. On subsequent logins, the username will <i>not</i> be added automatically to any groups. This change in behavior allows users to be removed from automatically-added groups. In Confluence 3.5 and 3.5.1, they would be re-added upon next login.</p> <p>Please note that there is no validation of the group names. If you mis-type the group name, authorization failures will result users will not be able to access the applications or functionality based on the intended group name.</p> <p>Examples:</p> <ul style="list-style-type: none"> <li>confluence-users</li> <li>confluence-users, jira-administrators, jira-core-users</li> </ul>

### Advanced settings

Setting	Description
---------	-------------

Enable Nested Groups	Enable or disable support for nested groups. Some directory servers allow you to define a group as a member of another group. Groups in such a structure are called <i>nested groups</i> . Nested groups simplify permissions by allowing sub-groups to inherit permissions from a parent group.
Manage User Status Locally	If true, you can activate and deactivate users in Crowd independent of their status in the directory server.
Filter out expired users	If true, user accounts marked as expired in ActiveDirectory will be automatically removed. For cached directories, the removal of a user will occur during the first synchronization after the account's expiration date.  <b>Note:</b> This is available in Embedded Crowd 2.0.0 and above, but not available in the 2.0.0 m04 release.
Use Paged Results	Enable or disable the use of the LDAP control extension for simple paging of search results. If paging is enabled, the search will retrieve sets of data rather than all of the search results at once. Enter the desired page size that is, the maximum number of search results to be returned per page when paged results are enabled. The default is 1000 results.
Follow Referrals	Choose whether to allow the directory server to redirect requests to other servers. This option uses the node referral (JNDI lookup <code>java.naming.referral</code> ) configuration setting. It is generally needed for Active Directory servers configured without proper DNS, to prevent a 'javax.naming.PartialResultException: Unprocessed Continuation Reference(s)' error.
Naive DN Matching	If your directory server will always return a consistent string representation of a DN, you can enable naive DN matching. Using naive DN matching will result in a significant performance improvement, so we recommend enabling it where possible.  This setting determines how your application will compare DN's to determine if they are equal. <ul style="list-style-type: none"> <li>• If this checkbox is selected, the application will do a direct, case-insensitive, string comparison. This is the default and recommended setting for Active Directory, because Active Directory guarantees the format of DN's.</li> <li>• If this checkbox is not selected, the application will parse the DN and then check the parsed version.</li> </ul>
Enable Incremental Synchronization	Enable incremental synchronization if you only want changes since the last synchronization to be queried when synchronizing a directory.  <b>⚠</b> Please be aware that when using this option, the user account configured for synchronization must have read access to: <ul style="list-style-type: none"> <li>• The <code>uSNChanged</code> attribute of all users and groups in the directory that need to be synchronized.</li> <li>• The objects and attributes in the Active Directory deleted objects container.</li> </ul> <p>If at least one of these conditions is not met, you may end up with users who are added to (or deleted from) the Active Directory not being respectively added (or deleted) in the application.</p> <p>This setting is only available if the directory type is set to "Microsoft Active Directory".</p>
Synchronization Interval (minutes)	Synchronization is the process by which the application updates its internal store of user data to agree with the data on the directory server. The application will send a request to your directory server every x minutes, where 'x' is the number specified here. The default value is 60 minutes.
Read Timeout (seconds)	The time, in seconds, to wait for a response to be received. If there is no response within the specified time period, the read attempt will be aborted. A value of 0 (zero) means there is no limit. The default value is 120 seconds.

Search Timeout (seconds)	The time, in seconds, to wait for a response from a search operation. A value of 0 (zero) means there is no limit. The default value is 60 seconds.
Connection Timeout (seconds)	This setting affects two actions. The default value is 0. <ul style="list-style-type: none"> <li>The time to wait when getting a connection from the connection pool. A value of 0 (zero) means there is no limit, so wait indefinitely.</li> <li>The time, in seconds, to wait when opening new server connections. A value of 0 (zero) means that the TCP network timeout will be used, which may be several minutes.</li> </ul>

## User schema settings

Setting	Description
User Object Class	This is the name of the class used for the LDAP user object. Example: <ul style="list-style-type: none"> <li><code>user</code></li> </ul>
User Object Filter	The filter to use when searching user objects. Example: <ul style="list-style-type: none"> <li><code>(&amp;(objectCategory=Person)(sAMAccountName=*))</code></li> </ul> <p>More examples can be found in our knowledge base. See <a href="#">How to write LDAP search filters</a>.</p>
User Name Attribute	The attribute field to use when loading the username. Examples: <ul style="list-style-type: none"> <li><code>cn</code></li> <li><code>sAMAccountName</code></li> </ul> <p>NB: In Active Directory, the 'sAMAccountName' is the 'User Logon Name (pre-Windows 2000)' field. The User Logon Name field is referenced by 'cn'.</p>
User Name RDN Attribute	The RDN (relative distinguished name) to use when loading the username. The DN for each LDAP entry is composed of two parts: the RDN and the location within the LDAP directory where the record resides. The RDN is the portion of your DN that is not related to the directory tree structure. Example: <ul style="list-style-type: none"> <li><code>cn</code></li> </ul>
User First Name Attribute	The attribute field to use when loading the user's first name. Example: <ul style="list-style-type: none"> <li><code>givenName</code></li> </ul>
User Last Name Attribute	The attribute field to use when loading the user's last name. Example: <ul style="list-style-type: none"> <li><code>sn</code></li> </ul>
User Display Name Attribute	The attribute field to use when loading the user's full name. Example: <ul style="list-style-type: none"> <li><code>displayName</code></li> </ul>

User Email Attribute	The attribute field to use when loading the user's email address. Example: <ul style="list-style-type: none"> <li>mail</li> </ul>
User Password Attribute	The attribute field to use when loading a user's password. Example: <ul style="list-style-type: none"> <li>unicodePwd</li> </ul>
User Unique ID Attribute	The attribute used as a unique immutable identifier for user objects. This is used to track username changes and is optional. If this attribute is not set (or is set to an invalid value), user renames will not be detected they will be interpreted as a user deletion then a new user addition.  This should normally point to a UUID value. Standards-compliant LDAP servers will implement this as 'entryUUID' according to <a href="#">RFC 4530</a> . This setting exists because it is known under different names on some servers, e.g. 'objectGUID' in Microsoft Active Directory.

### Group schema settings

Setting	Description
Group Object Class	This is the name of the class used for the LDAP group object. Examples: <ul style="list-style-type: none"> <li>groupOfUniqueNames</li> <li>group</li> </ul>
Group Object Filter	The filter to use when searching group objects. Example: <ul style="list-style-type: none"> <li>(&amp;(objectClass=group)(cn=*))</li> </ul>
Group Name Attribute	The attribute field to use when loading the group's name. Example: <ul style="list-style-type: none"> <li>cn</li> </ul>
Group Description Attribute	The attribute field to use when loading the group's description. Example: <ul style="list-style-type: none"> <li>description</li> </ul>

### Membership schema settings

Setting	Description
Group Members Attribute	The attribute field to use when loading the group's members. Example: <ul style="list-style-type: none"> <li>member</li> </ul>
User Membership Attribute	The attribute field to use when loading the user's groups. Example: <ul style="list-style-type: none"> <li>memberOf</li> </ul>

<p>Use the User Membership Attribute, when finding the user's group membership</p>	<p>Check this if your directory server supports the group membership attribute on the user. (By default, this is the 'memberOf' attribute.)</p> <ul style="list-style-type: none"> <li>• If this checkbox is selected, your application will use the group membership attribute on the user when <b>retrieving the list of groups to which a given user belongs</b>. This will result in a more efficient retrieval.</li> <li>• If this checkbox is not selected, your application will use the members attribute on the group ('member' by default) for the search.</li> <li>• If the <b>Enable Nested Groups</b> checkbox is selected, your application will ignore the <b>Use the User Membership Attribute</b> option and will use the members attribute on the group for the search.</li> </ul>
<p>Use the User Membership Attribute, when finding the members of a group</p>	<p>Check this if your directory server supports the user membership attribute on the group. (By default, this is the 'member' attribute.)</p> <ul style="list-style-type: none"> <li>• If this checkbox is selected, your application will use the group membership attribute on the user when <b>retrieving the members of a given group</b>. This will result in a more efficient search.</li> <li>• If this checkbox is not selected, your application will use the members attribute on the group ('member' by default) for the search.</li> </ul>

Diagrams of some possible configurations

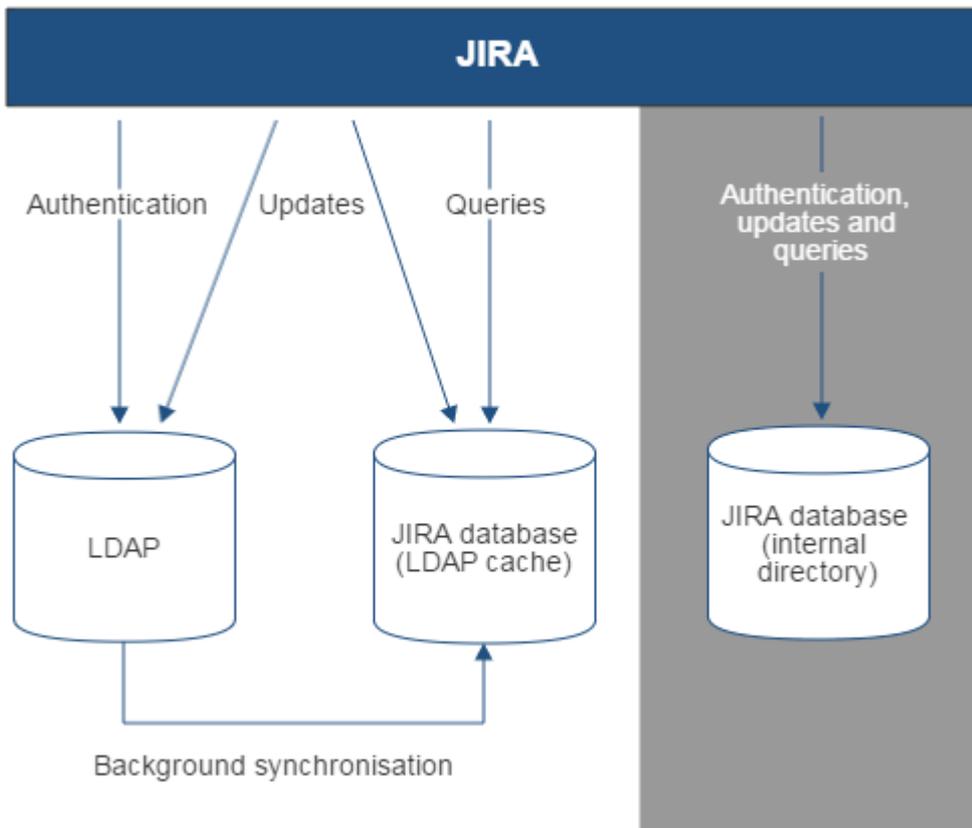


Diagram above: JIRA connecting to an LDAP directory.

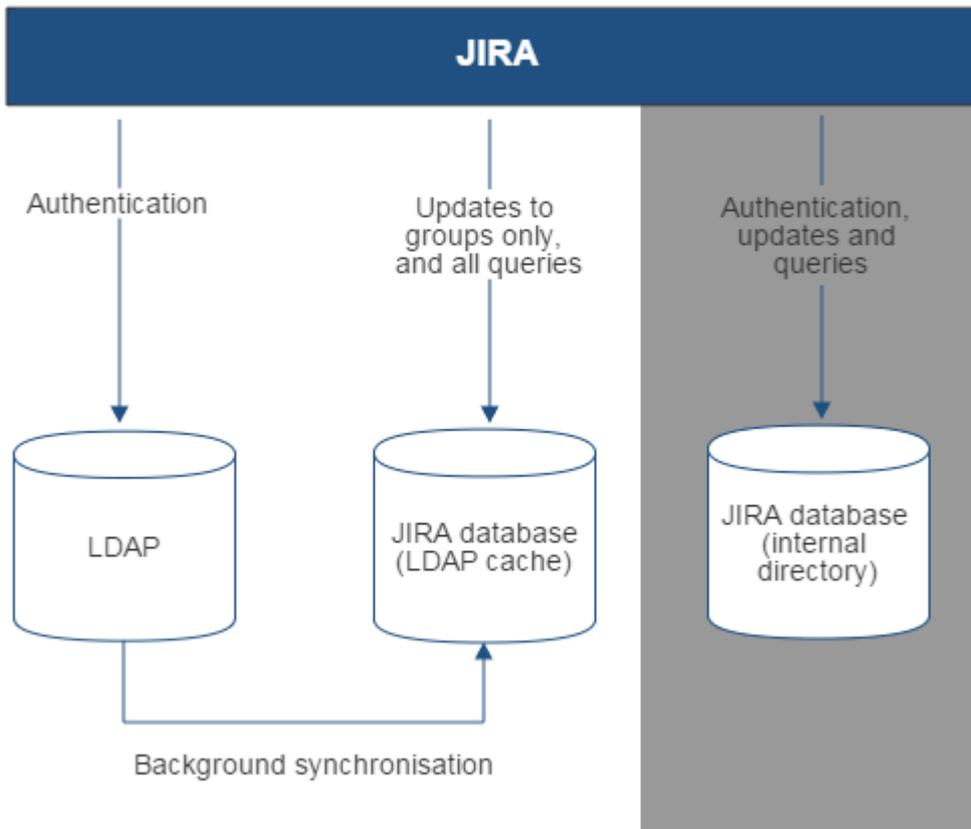


Diagram above: JIRA connecting to an LDAP directory with permissions set to read only and local groups.

Related topics

#### Configuring user directories

- [Configuring the internal directory](#)
- [Connecting to an LDAP directory](#)
  - [Configuring an SSL connection to Active Directory](#)
  - [Reducing the number of users synchronized from LDAP to JIRA applications](#)
- [Connecting to an internal directory with LDAP authentication](#)
- [Connecting to Crowd or another Jira application for user management](#)
- [Managing multiple directories](#)
- [Migrating users between user directories](#)
- [Synchronizing data from external directories](#)

# Configuring an SSL connection to Active Directory

 Atlassian applications allow the use of SSL within our applications, however Atlassian Support does not provide assistance for configuring it. Consequently, Atlassian **can not guarantee providing any support for it.**

- If assistance with conversions of certificates is required, please consult with the vendor who provided the certificate.
- If assistance with configuration is required, please raise a question on [Atlassian Answers](#).

If you want to configure a read/write connection with Microsoft Active Directory, you will need to install an SSL certificate, generated by your Active Directory server, onto your Jira server and then install the certificate into your JVM keystore.

## On this page:

- [Prerequisites](#)
- [Step 1. Install the Active Directory Certificate Services](#)
- [Step 2. Obtain the Server Certificate](#)
- [Step 3. Import the Server Certificate](#)

Updating user, group, and membership details in Active Directory requires that your Atlassian application be running in a JVM that trusts the AD server. To do this, we generate a certificate on the Active Directory server, then import it into Java's `keystore`.

## Prerequisites

To generate a certificate, you need the following components installed on the Windows Domain Controller to which you're connecting.

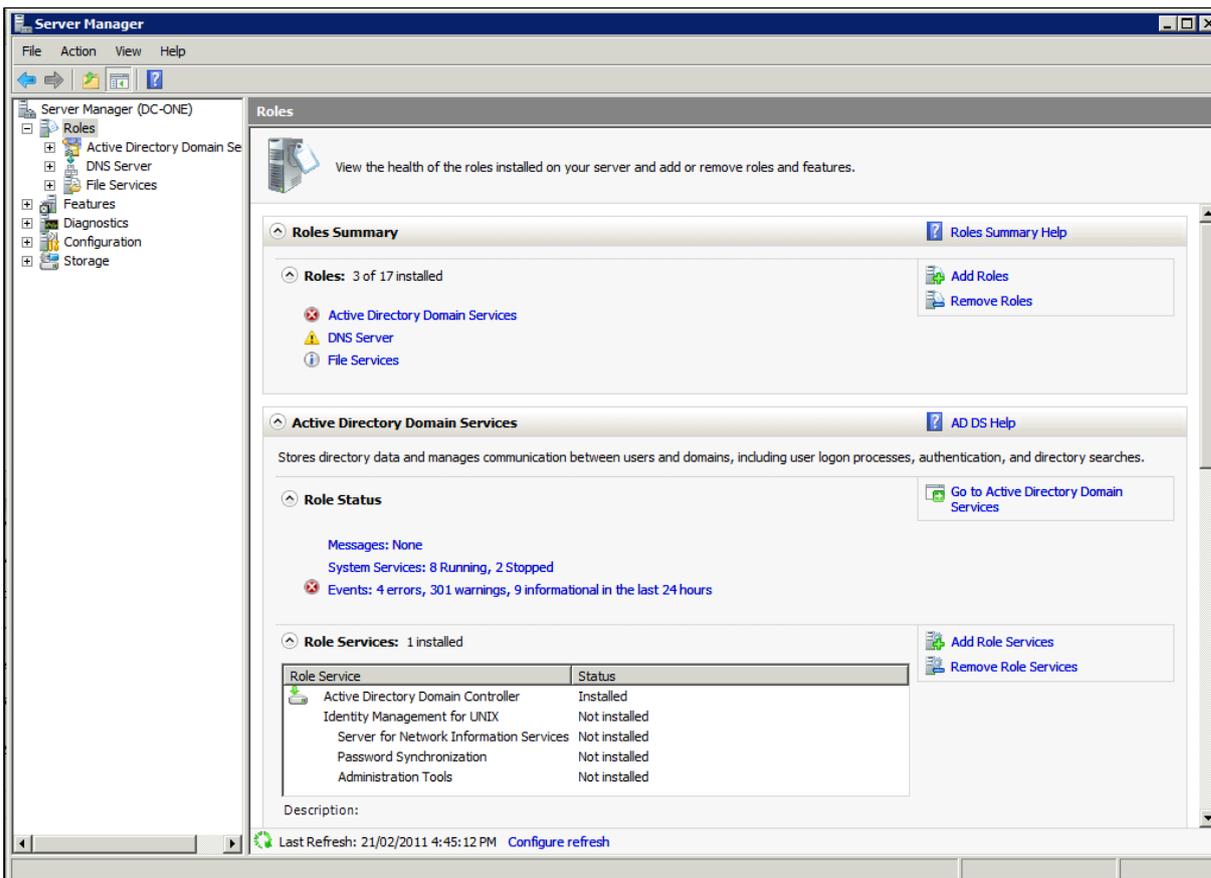
Required Component	Description
Internet Information Services (IIS)	This is required before you can install Windows Certificate Services.
Windows Certificate Services	This installs a certification authority (CA) which is used to issue certificates. Step 1, below, explains this process.
Windows 2000 Service Pack 2	Required if you are using Windows 2000
Windows 2000 High Encryption Pack (128-bit)	Required if you are using Windows 2000. Provides the highest available encryption level (128-bit).

## Step 1. Install the Active Directory Certificate Services

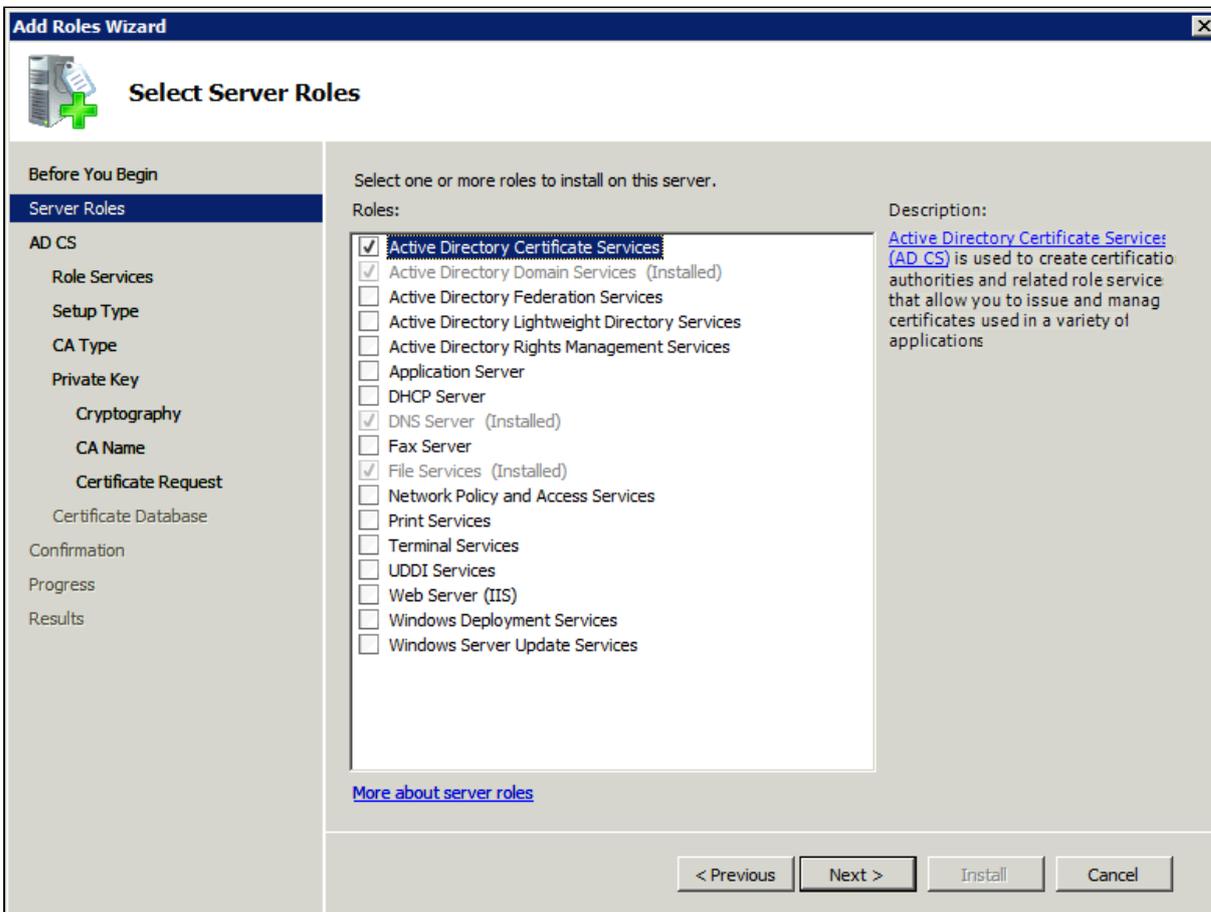
If Certificate Services are already installed, skip to step 2, below. The screenshots below are from Server 2008, but the process is similar for Server 2000 and 2003.

1. Log in to your Active Directory server as an administrator.
2. Click **Start**, point to **Administrative Tools**, and then click **Server Manager**.

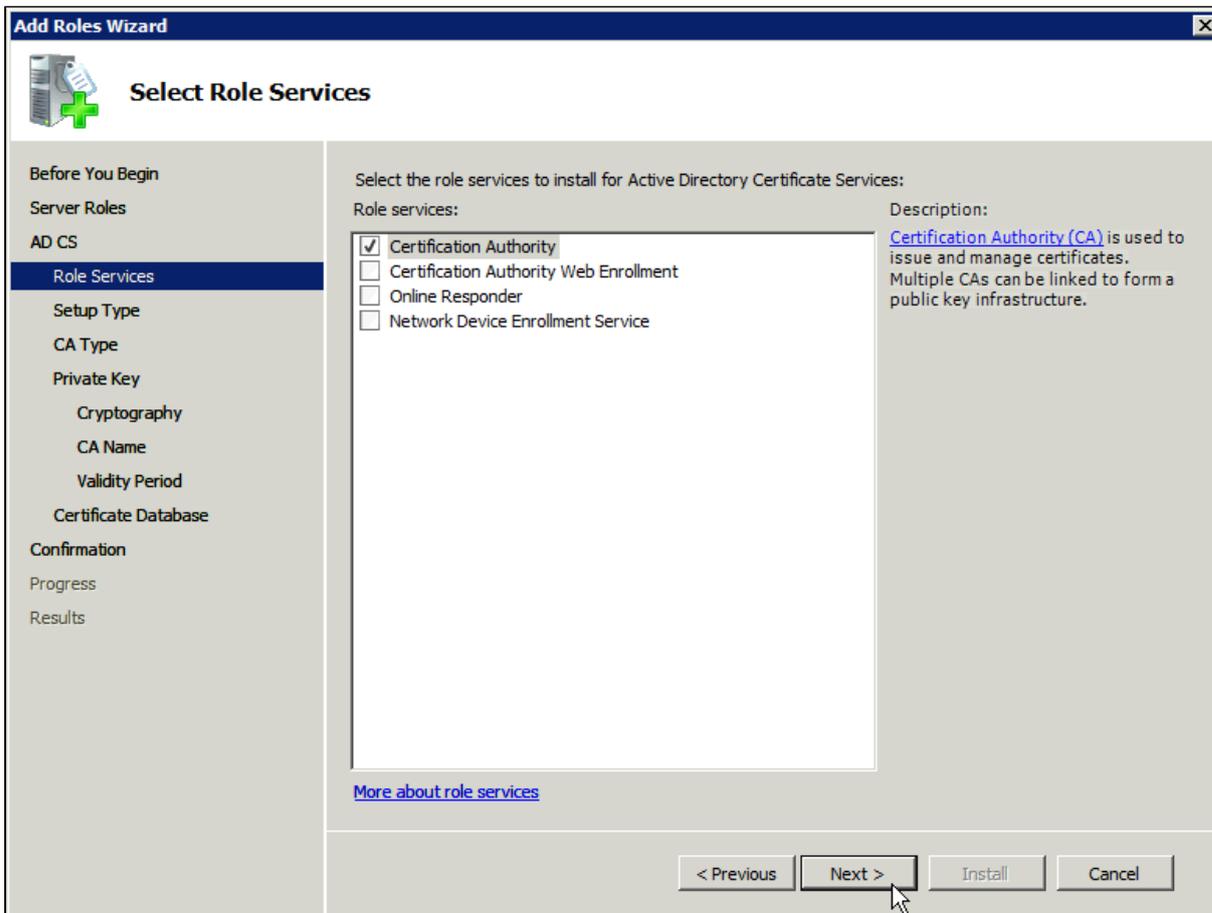
3. In the **ROLES Summary** section, click **Add Roles**.



4. On the **Select Server Roles** page, select the **Active Directory Certificate Services** check box. Click **Next** twice.



5. On the **Select Role Services** page, select the **Certification Authority** check box, and then click **Next**



6. On the **Specify Setup Type** page, click **Enterprise**, and then click **Next**.

**Add Roles Wizard**

**Specify Setup Type**

Before You Begin  
Server Roles  
AD CS  
Role Services  
**Setup Type**  
CA Type  
Private Key  
Cryptography  
CA Name  
Validity Period  
Certificate Database  
Confirmation  
Progress  
Results

Certification Authorities can use data in Active Directory to simplify the issuance and management of certificates. Specify whether you want to set up an Enterprise or Standalone CA.

Enterprise  
Select this option if this CA is a member of a domain and can use Directory Service to issue and manage certificates.

Standalone  
Select this option if this CA does not use Directory Service data to issue or manage certificates. A standalone CA can be a member of a domain.

[More about the differences between enterprise and standalone setup](#)

< Previous   Next >   Install   Cancel

7. On the **Specify CA Type** page, click **Root CA**, and then click **Next**.

**Add Roles Wizard**

**Specify CA Type**

Before You Begin  
Server Roles  
AD CS  
Role Services  
Setup Type  
**CA Type**  
Private Key  
Cryptography  
CA Name  
Validity Period  
Certificate Database  
Confirmation  
Progress  
Results

A combination of root and subordinate CAs can be configured to create a hierarchical public key infrastructure (PKI). A root CA is a CA that issues its own self-signed certificate. A subordinate CA receives its certificate from another CA. Specify whether you want to set up a root or subordinate CA.

Root CA  
Select this option if you are installing the first or only certification authority in a public key infrastructure.

Subordinate CA  
Select this option if your CA will obtain its CA certificate from another CA higher in a public key infrastructure.

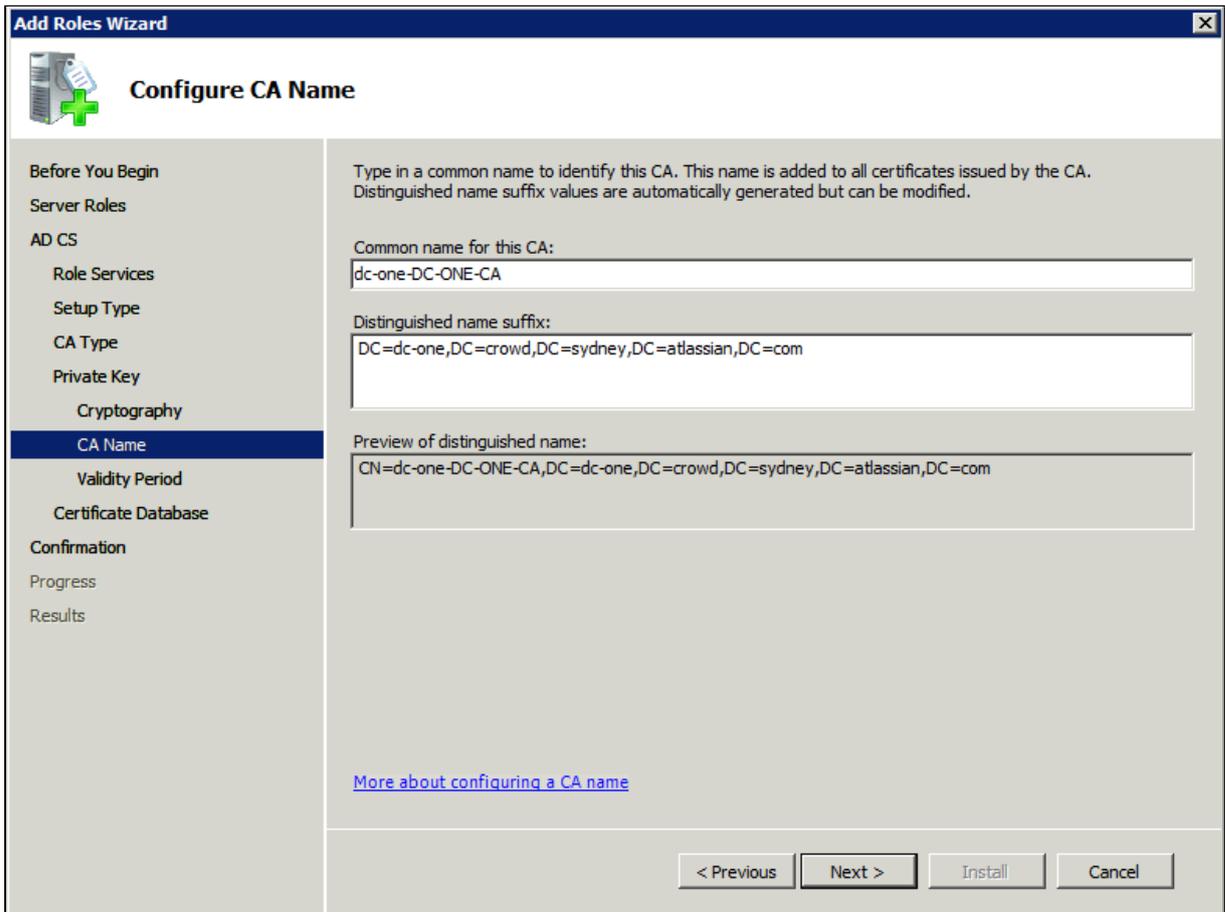
[More about public key infrastructure \(PKI\)](#)

< Previous   Next >   Install   Cancel

8. On the **Set Up Private Key** and **Configure Cryptography for CA** pages, you can configure optional configuration settings, including cryptographic service providers. However, the default values should be fine. Click **Next** twice.

The screenshot shows a window titled "Add Roles Wizard" with a sub-header "Set Up Private Key". On the left is a navigation pane with the following items: "Before You Begin", "Server Roles", "AD CS", "Role Services", "Setup Type", "CA Type", "Private Key" (highlighted), "Cryptography", "CA Name", "Validity Period", "Certificate Database", "Confirmation", "Progress", and "Results". The main content area contains the following text: "To generate and issue certificates to clients, a CA must have a private key. Specify whether you want to create a new private key or use an existing one." Below this are four radio button options: "Create a new private key" (selected), "Use existing private key", "Select a certificate and use its associated private key" (selected), and "Select an existing private key on this computer". Each option has a descriptive paragraph. At the bottom right are four buttons: "< Previous", "Next >", "Install", and "Cancel". A link "More about public and private keys" is located at the bottom left of the main content area.

- In the **Common name for this CA** box, type the common name of the CA, and then click **Next**.



The screenshot shows the 'Add Roles Wizard' dialog box with the 'Configure CA Name' step selected. The left sidebar contains a list of steps: 'Before You Begin', 'Server Roles', 'AD CS', 'Role Services', 'Setup Type', 'CA Type', 'Private Key', 'Cryptography', 'CA Name' (highlighted), 'Validity Period', 'Certificate Database', 'Confirmation', 'Progress', and 'Results'. The main area contains the following text and input fields:

Type in a common name to identify this CA. This name is added to all certificates issued by the CA. Distinguished name suffix values are automatically generated but can be modified.

Common name for this CA:

Distinguished name suffix:

Preview of distinguished name:

[More about configuring a CA name](#)

At the bottom right, there are four buttons: '< Previous', 'Next >', 'Install', and 'Cancel'.

- On the **Set Validity Period** page, accept the default values or specify other storage locations for the certificate database and the certificate database log, and then click **Next**.

**Add Roles Wizard**

### Set Validity Period

Before You Begin  
 Server Roles  
 AD CS  
 Role Services  
 Setup Type  
 CA Type  
 Private Key  
 Cryptography  
 CA Name  
**Validity Period**  
 Certificate Database  
 Confirmation  
 Progress  
 Results

A certificate will be issued to this CA to secure communications with other CAs and with clients requesting certificates. The validity period of a CA certificate can be based on a number of factors, including the intended purpose of the CA and security measures that you have taken to secure the CA.

Select validity period for the certificate generated for this CA:  
 Years

CA expiration Date: 21/02/2016 4:57 PM  
 Note that CA will issue certificates valid only until its expiration date.

[More about setting the certificate validity period](#)

< Previous   Next >   Install   Cancel

**Add Roles Wizard**

### Configure Certificate Database

Before You Begin  
 Server Roles  
 AD CS  
 Role Services  
 Setup Type  
 CA Type  
 Private Key  
 Cryptography  
 CA Name  
 Validity Period  
**Certificate Database**  
 Confirmation  
 Progress  
 Results

The certificate database records all certificate requests, issued certificates, and revoked or expired certificates. The database log can be used to monitor management activity for a CA.

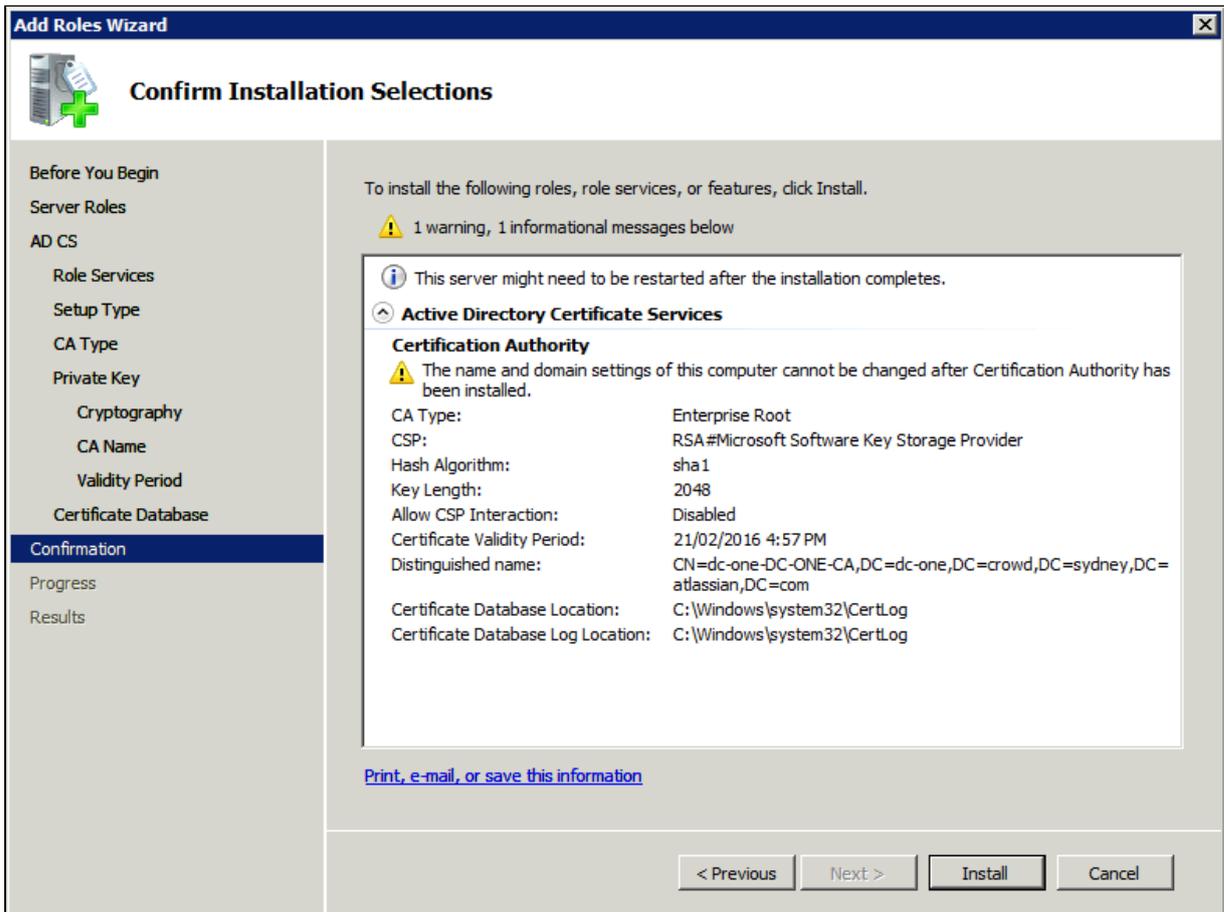
Certificate database location:  
 Browse...

Use existing certificate database from previous installation at this location

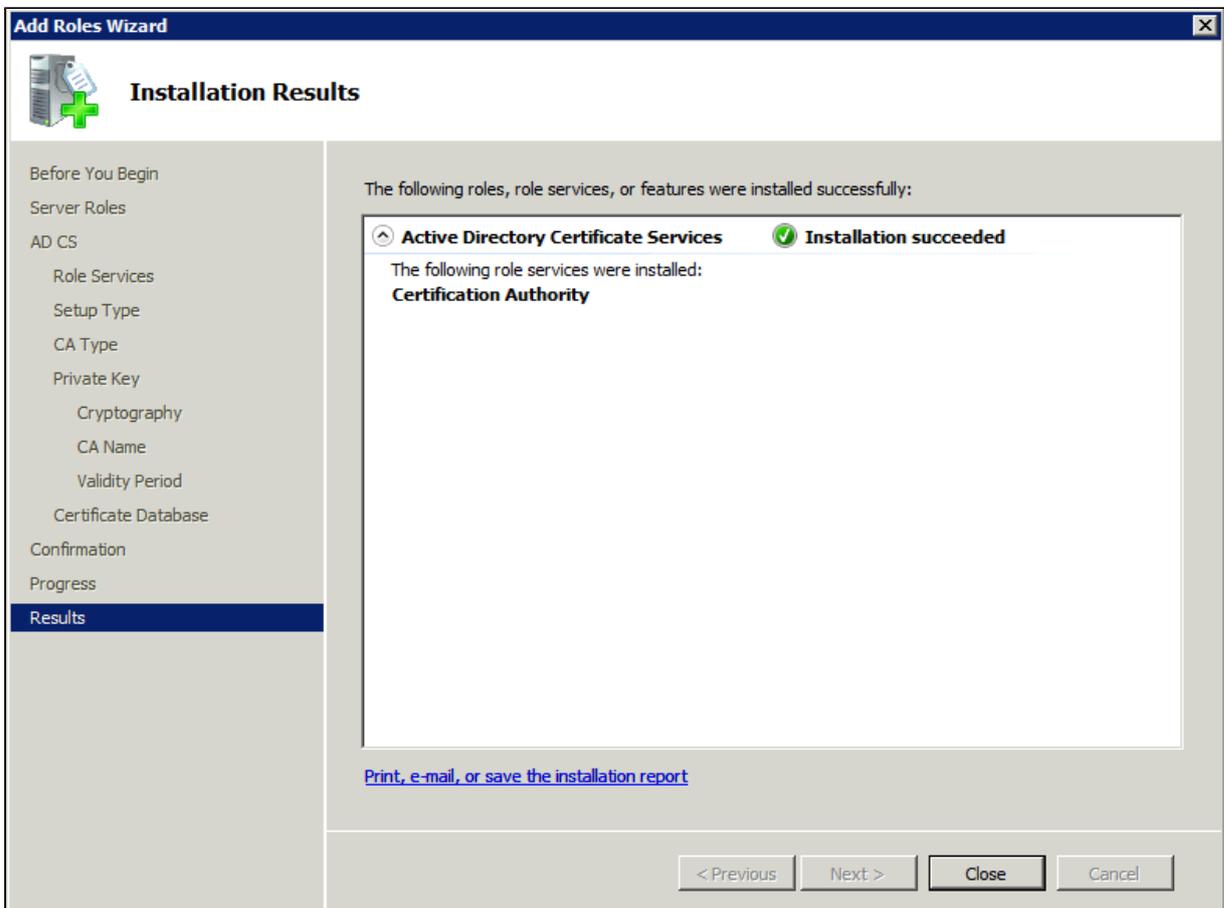
Certificate database log location:  
 Browse...

< Previous   Next >   Install   Cancel

11. After verifying the information on the **Confirm Installation Selections** page, click **Install**.



12. Review the information on the results screen to verify that the installation was successful.



## Step 2. Obtain the Server Certificate

The steps above describe how to install the certification authority (CA) on your Microsoft Active Directory server. Next, you will need to add the Microsoft Active Directory server's SSL certificate to the list of accepted certificates used by the JDK that runs your application server.

The Active Directory certificate is automatically generated and placed in root of the C:\ drive, matching a file format similar to the tree structure of your Active Directory server. For example: c:\ad2008.ad01.atlassian.com\_ad01.crt.

You can also export the certificate by executing this command on the Active Directory server:

```
certutil -ca.cert client.crt
```

You might still fail to be authenticated using the certificate file above. In this case, Microsoft's [LDAP over SSL \(LDAPS\) Certificate](#) page might help. Note that you need to:

1. Choose "No, do not export the private key" in step-10 of [Exporting the LDAPS Certificate and Importing for use with AD DS](#) section
2. Choose "DER encoded binary X.509 (.CER)" in step-11 of [Exporting the LDAPS Certificate and Importing for use with AD DS](#) section. This file will be used in the following step.

## Step 3. Import the Server Certificate

For an application server to trust your directory's certificate, the certificate must be imported into your Java runtime environment. The JDK stores trusted certificates in a file called a keystore. The default keystore file is called cacerts and it lives in the jre\lib\security sub-directory of your Java installation.

In the following examples, we use server-certificate.crt to represent the certificate file exported by your directory server. You will need to alter the instructions below to match the name actually generated.

Once the certificate has been imported as per the below instructions, you will need to restart the application to pick up the changes.

### Windows

1. Navigate to the directory in which Java is installed. It's probably called something like C:\Program Files\Java\jdk1.5.0\_12.

```
cd /d C:\Program Files\Java\jdk1.5.0_12
```

2. Run the command below, where server-certificate.crt is the name of the file from your directory server:

```
keytool -importcert -keystore .\jre\lib\security\cacerts -file server-certificate.crt
```

3. keytool will prompt you for a password. The default keystore password is changeit.
4. When prompted Trust this certificate? [no]: enter yes to confirm the key import:

```
Enter keystore password:  changeit
Owner: CN=ad01, C=US
Issuer: CN=ad01, C=US
Serial number: 15563d6677a4e9e4582d8a84be683f9
Valid from: Tue Aug 21 01:10:46 ACT 2007 until: Tue Aug 21 01:13:59 ACT 2012
Certificate fingerprints:
    MD5:  D6:56:F0:23:16:E3:62:2C:6F:8A:0A:37:30:A1:84:BE
    SHA1: 73:73:4E:A6:A0:D1:4E:F4:F3:CD:CE:BE:96:80:35:D2:B4:7C:79:C1
Trust this certificate? [no]:  yes
Certificate was added to keystore
```

5. Restart the application to take up the cacerts changes.

6. You may now change 'URL' to use LDAP over SSL (i.e. ldaps://<HOSTNAME>:636/) and use the 'Secure SSL' option when connecting your application to your directory server.

## UNIX

1. Navigate to the directory in which the Java used by JIRA is installed. If the default JAVA installation is used, then it would be

```
cd $JAVA_HOME
```

2. Run the command below, where server-certificate.crt is the name of the file from your directory server:

```
sudo keytool -importcert -keystore ./jre/lib/security/cacerts -file server-certificate.crt
```

3. keytool will prompt you for a password. The default keystore password is changeit.
4. When prompted Trust this certificate? [no]: enter yes to confirm the key import:

```
Password:
Enter keystore password:  changeit
Owner: CN=ad01, C=US
Issuer: CN=ad01, C=US
Serial number: 15563d6677a4e9e4582d8a84be683f9
Valid from: Tue Aug 21 01:10:46 ACT 2007 until: Tue Aug 21 01:13:59 ACT 2012
Certificate fingerprints:
    MD5:  D6:56:F0:23:16:E3:62:2C:6F:8A:0A:37:30:A1:84:BE
    SHA1: 73:73:4E:A6:A0:D1:4E:F4:F3:CD:CE:BE:96:80:35:D2:B4:7C:79:C1
Trust this certificate? [no]:  yes
Certificate was added to keystore
```

5. Restart the application to take up the cacerts changes.
6. You may now change 'URL' to use LDAP over SSL (i.e. ldaps://<HOSTNAME>:636/) and use the 'Secure SSL' option when connecting your application to your directory server.

## Mac OS X

1. Navigate to the directory in which Java is installed. This is usually

```
cd /Library/Java/Home
```

2. Run the command below, where server-certificate.crt is the name of the file from your directory server:

```
sudo keytool -importcert -keystore ./jre/lib/security/cacerts -file server-certificate.crt
```

3. keytool will prompt you for a password. The default keystore password is changeit.
4. When prompted Trust this certificate? [no]: enter yes to confirm the key import:

```
Password:
Enter keystore password:  changeit
Owner: CN=ad01, C=US
Issuer: CN=ad01, C=US
Serial number: 15563d6677a4e9e4582d8a84be683f9
Valid from: Tue Aug 21 01:10:46 ACT 2007 until: Tue Aug 21 01:13:59 ACT 2012
Certificate fingerprints:
    MD5:  D6:56:F0:23:16:E3:62:2C:6F:8A:0A:37:30:A1:84:BE
    SHA1: 73:73:4E:A6:A0:D1:4E:F4:F3:CD:CE:BE:96:80:35:D2:B4:7C:79:C1
Trust this certificate? [no]:  yes
Certificate was added to keystore
```

5. Restart the application to take up the cacerts changes.
6. You may now change 'URL' to use LDAP over SSL (i.e. ldaps://<HOSTNAME>:636/) and use the 'Secure SSL' option when connecting your application to your directory server.

Related topics

[Connecting to an LDAP directory](#)  
[Configuring user directories](#)

# Reducing the number of users synchronized from LDAP to JIRA applications

If you have [connected Jira applications to an LDAP directory](#) for authentication, user and group management, you may want configure your applications to synchronize a subset of users from LDAP rather than all users. There are two reasons for why you might make this change:

- **Improving performance** If you have performance issues during synchronization process, you may be able to improve this by synchronizing a subset of data instead. See this knowledge base article for more information: [Performance issues with large LDAP repository in Jira server](#).
- **Reducing your user count** You can synchronize a subset of users to Jira applications from LDAP to reduce your user count. This will allow you to count less users against your Jira application licenses. However, synchronizing a subset of users to Jira applications from LDAP is not the recommended method for reducing your Jira application user count. We recommend that you reduce the Jira application user count by deactivating the users within Jira. Check [this page](#) for more info on removing users from Jira.

## Procedure

The procedure for configuring Jira applications to synchronize a different number of users from LDAP depends on how you initially set up your LDAP directory. For example, if you have all your Jira application users in one organizational unit and your non-Jira application users in another organizational unit, then you can simply configure Jira applications to only synchronize users against a particular DN (distinguished name). However, if your setup is not so simple (e.g. you have your Jira application users and non-Jira application users in the same node), you will need to define an LDAP filter to synchronize the relevant users. Both of these methods are outlined below.

### Synchronizing against Base DN, Additional User DN and Additional Group DN

1. Log in as a user with the [Jira Administrators global permission](#).
2. Select **Administration > Users > User Directories**.
3. Update the **Base DN** field, and optionally the **Additional User DN** and/or **Additional Group DN** to query against the directory server as desired.
4. For example, if you have configured all of your Jira application users in the jira-users organizational unit only, for your company at mycompany.example.com, your configuration would look like this:

- **Base DN** `dc=mycompany,dc=example,dc=com`
- **Additional Group DN** `ou=jira-users`

### Defining an LDAP filter

1. Log in as a user with the [Jira Administrators global permission](#).  
Select **Administration > Users > User Directories**.
2. Update **User Object Filter** and/or **Group Object Filter** fields as desired. The syntax for LDAP filters is not simple and your query will depend on how you have set up your LDAP directory.
3. For example, if you have configured only Jira application groups to have 'jira' in the CN, you can use a wildcard search in your filter to find them by setting the **Group Object Filter** = `(objectCategory=group)(cn=*jira*)`  
More information on defining LDAP filters is available in the pages linked in the *Related Topics* section below.

---

## Related topics:

[Performance issues with large LDAP repository in Jira server](#)

[Unable to create issues due to exceeded number of licenses](#)

[How to write LDAP search filters](#)

[MSDN guide to LDAP search filter syntax](#)

# Connecting to an internal directory with LDAP authentication

You can connect your Jira application to an LDAP directory for delegated authentication. This means that Jira will have an internal directory that uses LDAP for authentication only. There is an option to create users in the internal directory automatically when they attempt to log in, as described in the settings section.

 If you decide to use an LDAP directory for delegated authentication, you're unable to use [nested groups](#).

You will need to log in as a user with the 'Jira System Administrators' [global permission](#) to access the Settings menu below.

## On this page:

- [Overview](#)
- [Connecting Jira to an internal directory with LDAP authentication](#)
- [Server settings](#)
  - [Copying users on first login](#)
- [Schema settings](#)
- [User schema settings \(used when copying users on first login\)](#)
- [Group schema settings \(used when enabling 'synchronize group memberships'\)](#)
- [Diagrams of possible configurations](#)

 Managing 500+ users across Atlassian products? Find out how easy, scalable and effective it can be with Crowd! See [centralized user management](#).

## Overview

An internal directory with LDAP authentication offers the features of an internal directory while allowing you to store and check users' passwords in LDAP only. Note that the 'internal directory with LDAP authentication' is separate from the default 'internal directory'. On LDAP, all that the application does is to check the password. The LDAP connection is read only. Every user in the internal directory with LDAP authentication must map to a user on LDAP, otherwise they cannot log in.

**When to use this option:** Choose this option if you want to set up a user and group configuration within your application that suits your needs, while checking your users' passwords against the corporate LDAP directory. This option also helps to avoid the performance issues that may result from downloading large numbers of groups from LDAP.

## Connecting Jira to an internal directory with LDAP authentication

### To connect to an internal directory but check logins via LDAP:

1. Choose **Administration** () > **User Management**.
2. Choose **User Directories**.
3. **Add** a directory and select type '**Internal with LDAP Authentication**'.
4. Enter the values for the settings, as described below.
5. Save the directory settings.

6. Define the **directory order** by clicking the blue up- and down-arrows next to each directory on the '**User Directories**' screen. We recommend that the 'Internal Directory with LDAP Authentication' is at the top of the list. Here is a summary of how the directory order affects the processing:

- The order of the directories is the order in which they will be searched for users and groups.
- Changes to users and groups will be made only in the first directory where the application has permission to make changes.

For details, see [Managing multiple directories](#).

7. Add your users and groups in Jira. See [Managing users](#) and [Managing groups](#).

## Server settings

Setting	Description
Name	A descriptive name that will help you to identify the directory. Examples: <ul style="list-style-type: none"> <li>• Internal directory with LDAP Authentication</li> <li>• Corporate LDAP for Authentication Only</li> </ul>
Directory Type	Select the type of LDAP directory that you will connect to. If you are adding a new LDAP connection, the value you select here will determine the default values for some of the options on the rest of screen. Examples: <ul style="list-style-type: none"> <li>• Microsoft Active Directory</li> <li>• OpenDS</li> <li>• And more.</li> </ul>
Host name	The host name of your directory server. Examples: <ul style="list-style-type: none"> <li>• ad.example.com</li> <li>• ldap.example.com</li> <li>• opens.example.com</li> </ul>
Port	The port on which your directory server is listening. Examples: <ul style="list-style-type: none"> <li>• 389</li> <li>• 10389</li> <li>• 636 (for example, for SSL)</li> </ul>
Use SSL	Check this box if the connection to the directory server is an SSL (Secure Sockets Layer) connection. Note that you will need to configure an SSL certificate in order to use this setting.
Username	The distinguished name of the user that the application will use when connecting to the directory server. Examples: <ul style="list-style-type: none"> <li>• cn=administrator,cn=users,dc=ad,dc=example,dc=com</li> <li>• cn=user,dc=domain,dc=name</li> <li>• user@domain.name</li> </ul>
Password	The password of the user specified above.

## Copying users on first login

Setting	Description
Copy User on Login	This option affects what will happen when a user attempts to log in. If this box is checked, the user will be created automatically in the internal directory that is using LDAP for authentication when the user first logs in and their details will be synchronized on each subsequent log in. If this box is not checked, the user's login will fail if the user wasn't already manually created in

	<p>the directory.</p> <p>If you check this box the following additional fields will appear on the screen, which are described in more detail below:</p> <ul style="list-style-type: none"> <li>• Default Group Memberships</li> <li>• Synchronize Group Memberships</li> <li>• User Schema Settings (described in a separate section below)</li> </ul>
Update User attributes on Login	<p>Whenever your users authenticate to the application, their attributes will be automatically updated from the LDAP server into the application. After you select this option, you won't be able to modify or delete your users directly in the application.</p> <ul style="list-style-type: none"> <li>• If you need to modify a user, do it on the LDAP server; it will be updated in the application after authenticating.</li> <li>• If you need to delete a user, do it on the LDAP server, but also in the application. If you delete the user only on the LDAP server, it will be rejected from logging in to the application, but it won't be set as inactive, which will affect your license. You'll need to disable the <b>Update User attributes on Login</b> option to delete the user, and then enable it again.</li> </ul>
Default Group Memberships	<p>This field appears if you check the <b>Copy User on Login</b> box. If you would like users to be automatically added to a group or groups, enter the group name(s) here. To specify more than one group, separate the group names with commas. Each time a user logs in, their group memberships will be checked. If the user does not belong to the specified group(s), their username will be added to the group(s). If a group does not yet exist, it will be added to the internal directory that is using LDAP for authentication.</p> <p>Please note that there is no validation of the group names. If you mis-type the group name, authorization failures will result users will not be able to access the applications or functionality based on the intended group name.</p> <p>Examples:</p> <ul style="list-style-type: none"> <li>• <code>confluence-users</code></li> <li>• <code>bamboo-users, jira-administrators, jira-core-users</code></li> </ul>
Synchronize Group Memberships	<p>This field appears if you select the <b>Copy User on Login</b> checkbox. If this box is checked, group memberships specified on your LDAP server will be synchronized with the internal directory each time the user logs in.</p> <p>If you check this box the following additional fields will appear on the screen, both described in more detail below:</p> <ul style="list-style-type: none"> <li>• Group Schema Settings (described in a separate section below)</li> <li>• Membership Schema Settings (described in a separate section below)</li> </ul>

## Schema settings

Setting	Description
Base DN	<p>The root distinguished name (DN) to use when running queries against the directory server.</p> <p>Examples:</p> <ul style="list-style-type: none"> <li>• <code>o=example, c=com</code></li> <li>• <code>cn=users, dc=ad, dc=example, dc=com</code></li> </ul>

	<ul style="list-style-type: none"> <li>For Microsoft Active Directory, specify the base DN in the following format: <code>dc=domain1,dc=local</code>. You will need to replace the <code>domain1</code> and <code>local</code> for your specific configuration. Microsoft Server provides a tool called <code>ldp.exe</code> which is useful for finding out and configuring the the LDAP structure of your server.</li> </ul>
User Name Attribute	<p>The attribute field to use when loading the username. Examples:</p> <ul style="list-style-type: none"> <li><code>cn</code></li> <li><code>sAMAccountName</code></li> </ul>

### User schema settings (used when copying users on first login)

Setting	Description
Additional User DN	<p>This value is used in addition to the base DN when searching and loading users. If no value is supplied, the subtree search will start from the base DN. Example:</p> <ul style="list-style-type: none"> <li><code>ou=Users</code></li> </ul>
User Object Class	<p>This is the name of the class used for the LDAP user object. Example:</p> <ul style="list-style-type: none"> <li><code>user</code></li> </ul>
User Object Filter	<p>The filter to use when searching user objects. Example:</p> <ul style="list-style-type: none"> <li><code>(&amp;(objectCategory=Person)(sAMAccountName=*))</code></li> </ul>
User Name RDN Attribute	<p>The RDN (relative distinguished name) to use when loading the username. The DN for each LDAP entry is composed of two parts: the RDN and the location within the LDAP directory where the record resides. The RDN is the portion of your DN that is not related to the directory tree structure. Example:</p> <ul style="list-style-type: none"> <li><code>cn</code></li> </ul>
User First Name Attribute	<p>The attribute field to use when loading the user's first name. Example:</p> <ul style="list-style-type: none"> <li><code>givenName</code></li> </ul>
User Last Name Attribute	<p>The attribute field to use when loading the user's last name. Example:</p> <ul style="list-style-type: none"> <li><code>sn</code></li> </ul>
User Display Name Attribute	<p>The attribute field to use when loading the user's full name. Example:</p> <ul style="list-style-type: none"> <li><code>displayName</code></li> </ul>
User Email Attribute	<p>The attribute field to use when loading the user's email address. Example:</p> <ul style="list-style-type: none"> <li><code>mail</code></li> </ul>

### Group schema settings (used when enabling 'synchronize group memberships')

Setting	Description
---------	-------------

Group Object Class	This is the name of the class used for the LDAP group object. Examples: <ul style="list-style-type: none"> <li>• groupOfUniqueNames</li> <li>• group</li> </ul>
Group Object Filter	The filter to use when searching group objects. Example: <ul style="list-style-type: none"> <li>• (&amp;(objectClass=group)(cn=*))</li> </ul>
Group Name Attribute	The attribute field to use when loading the group's name. Example: <ul style="list-style-type: none"> <li>• cn</li> </ul>
Group Description Attribute	The attribute field to use when loading the group's description. Example: <ul style="list-style-type: none"> <li>• description</li> </ul>

### Diagrams of possible configurations

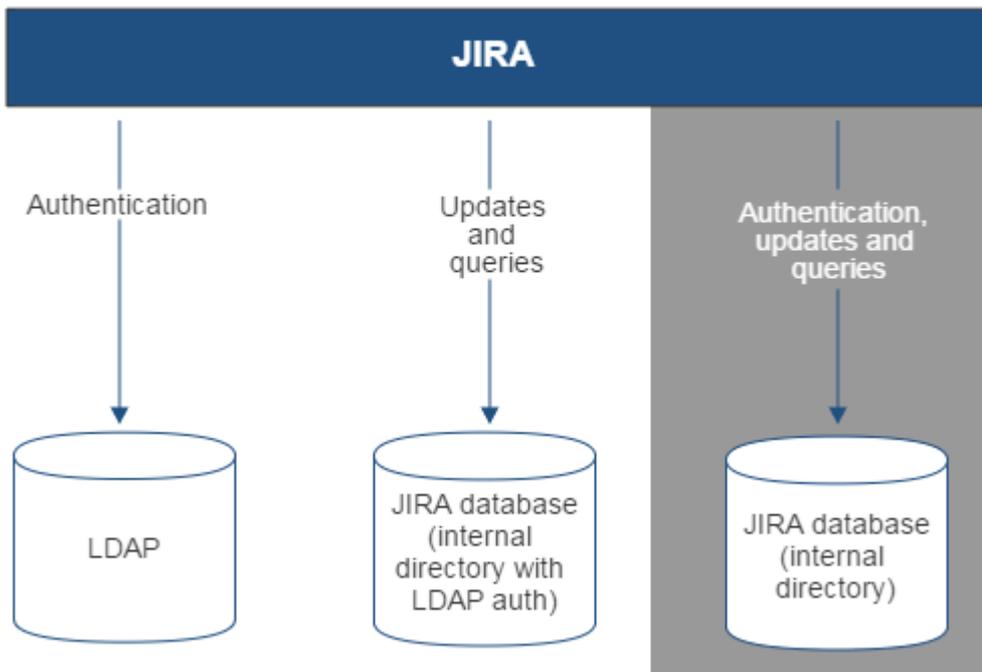


Diagram above: JIRA connecting to an LDAP directory for authentication only.

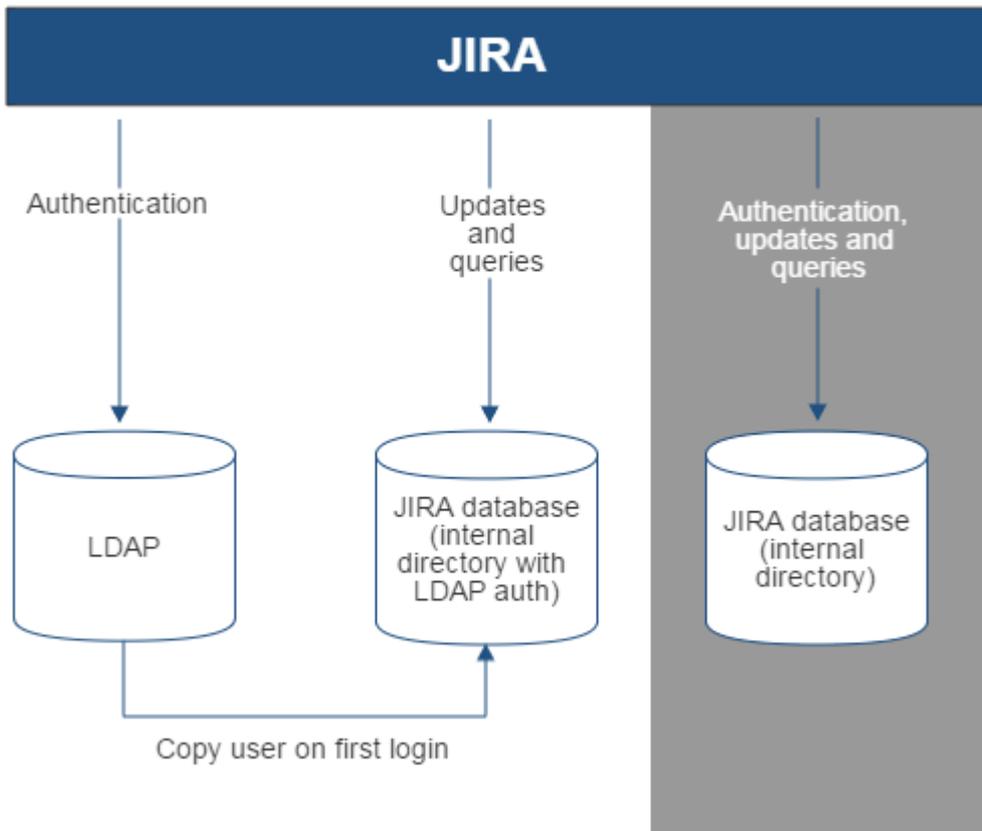


Diagram above: JIRA connecting to an LDAP directory for authentication only, with each user copied to the internal directory when they first log in to JIRA.

Related topics

#### Configuring user directories

- [Configuring the internal directory](#)
- [Connecting to an LDAP directory](#)
- [Connecting to an internal directory with LDAP authentication](#)
- [Connecting to Crowd or another Jira application for user management](#)
- [Managing multiple directories](#)
- [Migrating users between user directories](#)
- [Synchronizing data from external directories](#)

# Connecting to Crowd or another Jira application for user management

You can connect your Jira application to Atlassian Crowd or to another Jira server application (version 4.3 or later) for management of users and groups, and for authentication (verification of a user's login). You will need to log in as a user with the 'Jira System Administrators' [global permission](#) to access the Settings menu below.

## On this page:

- [Connecting a Jira application to Crowd](#)
- [Connecting Jira applications to another server](#)
- [Diagrams of some possible configurations](#)

## Connecting a Jira application to Crowd

Atlassian Crowd is an application security framework that handles authentication and authorization for your web-based applications. With Crowd you can integrate multiple web applications and user directories, with support for single sign-on (SSO) and centralized identity management. The Crowd Administration Console provides a web interface for managing directories, users and their permissions. See the [Administration Guide](#).

**When to use this option:** Connect to Crowd if you want to use the full Crowd functionality to manage your directories, users and groups. You can connect your Crowd server to a number of directories of all types that Crowd supports, including custom directory connectors.

### To connect a Jira application to Crowd:

1. Go to your **Crowd Administration Console** and define the Jira application to Crowd. See the Crowd documentation: [Adding an Application](#).
2. Choose **Administration**  > **User Management**.
3. Choose **User Directories**.
4. **Add** a directory and select type '**Atlassian Crowd**'. Enter the settings as described below.
5. Save the directory settings.
6. Define the **directory order** by clicking the blue up- and down-arrows next to each directory on the '**User Directories**' screen. Here is a summary of how the directory order affects the processing:
  - The order of the directories is the order in which they will be searched for users and groups.
  - Changes to users and groups will be made only in the first directory where the application has permission to make changes.For details, see [Managing multiple directories](#).
7. If required, configure Jira to use Crowd for single sign-on (SSO) too. See the Crowd documentation: [Integrating Crowd with Atlassian Jira](#).

### Notes:

- If you have Jira-Crowd-LDAP, every time user logs in (i.e. first and subsequent times), the user's data in Jira/Crowd will be updated from the user's data in LDAP. This includes username, display name, email and group memberships. However for group memberships, only the following applies:
  - direct groups only (i.e. not nested groups) are synchronized from LDAP.
  - only groups that are already present in the Jira application are synchronized, i.e. groups are not added/removed, and group hierarchies are not synchronized.

### Settings in Jira applications for the Crowd directory type

Setting	Description
---------	-------------

Name	A meaningful name that will help you to identify this Crowd server amongst your list of directory servers. Examples: <ul style="list-style-type: none"> <li>• Crowd Server</li> <li>• Example Company Crowd</li> </ul>
Server URL	The web address of your Crowd console server. Examples: <ul style="list-style-type: none"> <li>• <a href="http://www.example.com:8095/crowd/">http://www.example.com:8095/crowd/</a></li> <li>• <a href="http://crowd.example.com">http://crowd.example.com</a></li> </ul>
Application Name	The name of your application, as recognized by your Crowd server. Note that you will need to define the application in Crowd too, using the Crowd administration Console. See the Crowd documentation on <a href="#">adding an application</a> .
Application Password	The password which the application will use when it authenticates against the Crowd framework as a client. This must be the same as the password you have registered in Crowd for this application. See the Crowd documentation on <a href="#">adding an application</a> .

### Crowd permissions

Setting	Description
Read Only	The users, groups and memberships in this directory are retrieved from Crowd and can only be modified via Crowd. You cannot modify Crowd users, groups or memberships via the application administration screens.
Read /Write	The users, groups and memberships in this directory are retrieved from Crowd. When you modify a user, group or membership via the application administration screens, the changes will be applied directly to Crowd. Please ensure that the application has modification permissions for the relevant directories in Crowd. See the Crowd documentation: <a href="#">Specifying an Application's Directory Permissions</a> .

### Advanced Crowd settings

Setting	Description
Enable Nested Groups	Enable or disable support for nested groups. Before enabling nested groups, please check to see if the user directory or directories in Crowd support nested groups. When nested groups are enabled, you can define a group as a member of another group. If you are using groups to manage permissions, you can create nested groups to allow inheritance of permissions from one group to its sub-groups.
Enable Incremental Synchronization	Enable or disable incremental synchronization. Only changes since the last synchronization will be retrieved when synchronizing a directory. Note that full synchronization is always executed when restarting the application.
Synchronization Interval (minutes)	Synchronization is the process by which the application updates its internal store of user data to agree with the data on the directory server. The application will send a request to your directory server every x minutes, where 'x' is the number specified here. The default value is 60 minutes.

 Managing 500+ users across Atlassian products? Find out how easy, scalable and effective it can be with Crowd! See [centralized user management](#).

## Connecting Jira applications to another server

Subject to certain limitations, you can connect a number of Atlassian applications to a single JIRA application for centralized user management.

**When to use this option:** You can connect to a server running **Jira 4.3** or later, **Jira Software 7.0** or later, **Jira Core 7.0** or later, or **Jira Service Management (formerly Jira Service Desk) 3.0** or later. Choose this option as an alternative to Atlassian Crowd, for simple configurations with a limited number of users.

Let's assume that you have two Jira applicationservers, called for example '**Jira instance 1**' and '**Jira instance2**'. You want Jira instance2 to manage your users and groups. Jira instance1 will delegate user management to Jira instance2.

### To connect Jira instance1 to use Jira instance2 for user management:

1. Configure Jira instance2 to recognize Jira instance1:
  - Choose **Administration**  > **User Management**.
  - Choose **User Directories**.
  - **Add** an application.
  - Enter the **application name** and **password** that Jira instance1 will use when accessing Jira instance2.
  - Enter the **IP address** or addresses of Jira instance1. Valid values are:
    - A full IP address, e.g. 192.168.10.12.
    - A wildcard IP range, using CIDR notation, e.g. 192.168.10.1/16. For more information, see the introduction to [CIDR notation on Wikipedia](#) and [RFC 4632](#).
  - **Save** the new application.
2. Configure Jira instance1 to delegate user management:
  - Choose **Administration**  > **User Management**.
  - Choose **User Directories**.
  - **Add** a directory and select type '**Atlassian Jira**'.
  - Enter the settings as described below. When asked for the **application name** and **password**, enter the values that you defined in the settings on Jira instance2.
  - Save the directory settings.
  - Define the **directory order** by clicking the blue up- and down-arrows next to each directory on the '**User Directories**' screen. Here is a summary of how the directory order affects the processing:
    - The order of the directories is the order in which they will be searched for users and groups.
    - Changes to users and groups will be made only in the first directory where the application has permission to make changes.

For details, see [Managing multiple directories](#).

### Settings for the Jira application directory type

Setting	Description
Name	A meaningful name that will help you to identify this Jira server in the list of directory servers. Examples: <ul style="list-style-type: none"> <li>• Jira Software</li> <li>• My Company Jira</li> </ul>
Server URL	The web address of your Jira server. Examples: <ul style="list-style-type: none"> <li>• http://www.example.com:8080</li> <li>• http://jira.example.com</li> </ul>
Application Name	The name used by your application when accessing the Jira server that acts as user manager. Note that you will also need to define your application to that Jira server, via the ' <b>Other Applications</b> ' option in the 'Users, Groups & Roles' section of the 'Administration' menu.

Application Password	The password used by your application when accessing the Jira server that acts as user manager.
----------------------	---

### Permissions for the Jira applicationdirectory type

Setting	Description
Read Only	The users, groups and memberships in this directory are retrieved from the Jira server that is acting as user manager. They can only be modified via that JIRA server.

### Advanced Settings for the Jira applicationdirectory type

Setting	Description
Enable Nested Groups	Enable or disable support for nested groups. Before enabling nested groups, please check to see if nested groups are enabled on the JIRA server that is acting as user manager. When nested groups are enabled, you can define a group as a member of another group. If you are using groups to manage permissions, you can create nested groups to allow inheritance of permissions from one group to its sub-groups.
Synchronization Interval (minutes)	Synchronization is the process by which the application updates its internal store of user data to agree with the data on the directory server. The application will send a request to your directory server every x minutes, where 'x' is the number specified here. The default value is 60 minutes.

Diagrams of some possible configurations

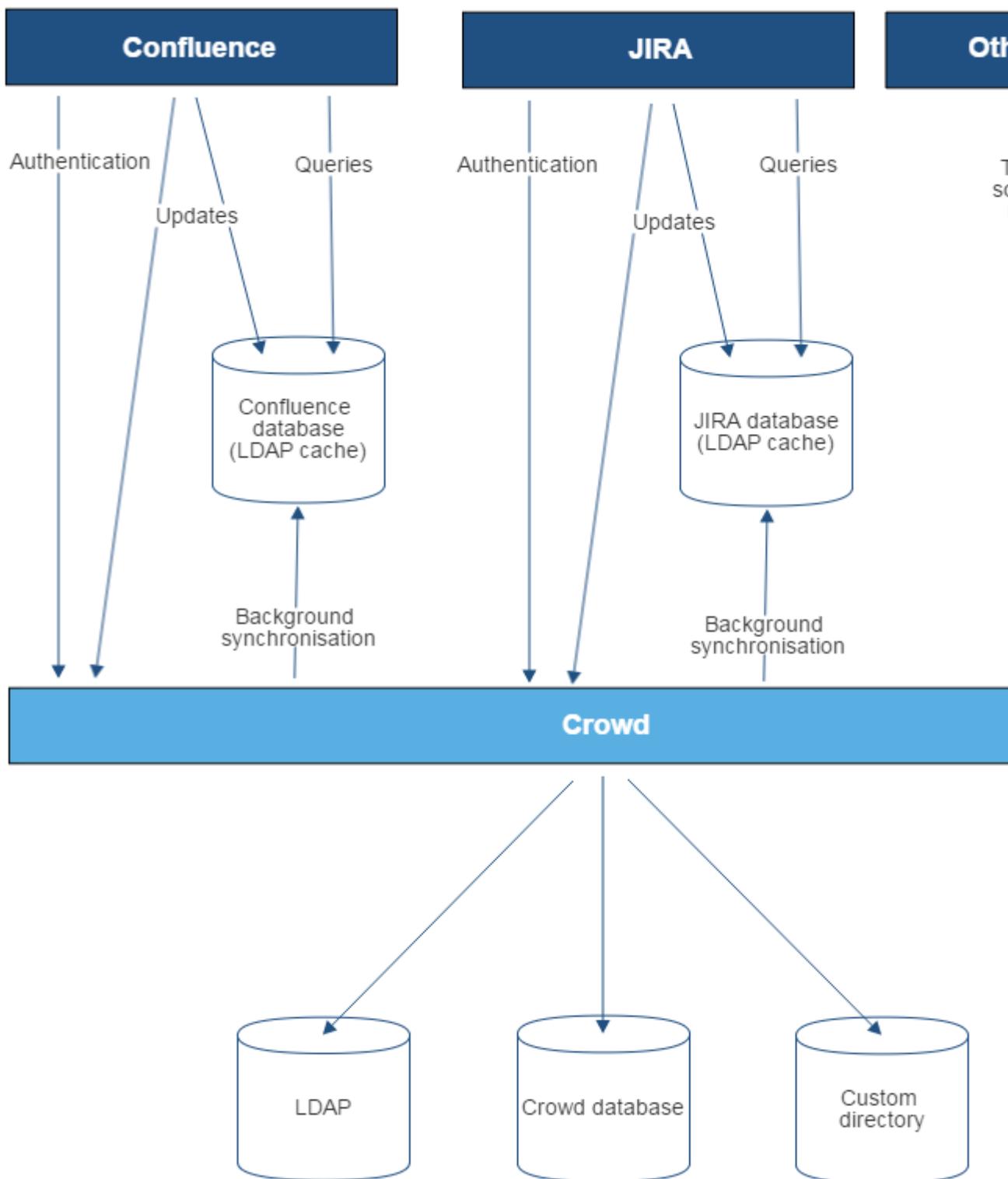


Diagram above: Confluence, JIRA and other applications connecting to Crowd for user management.

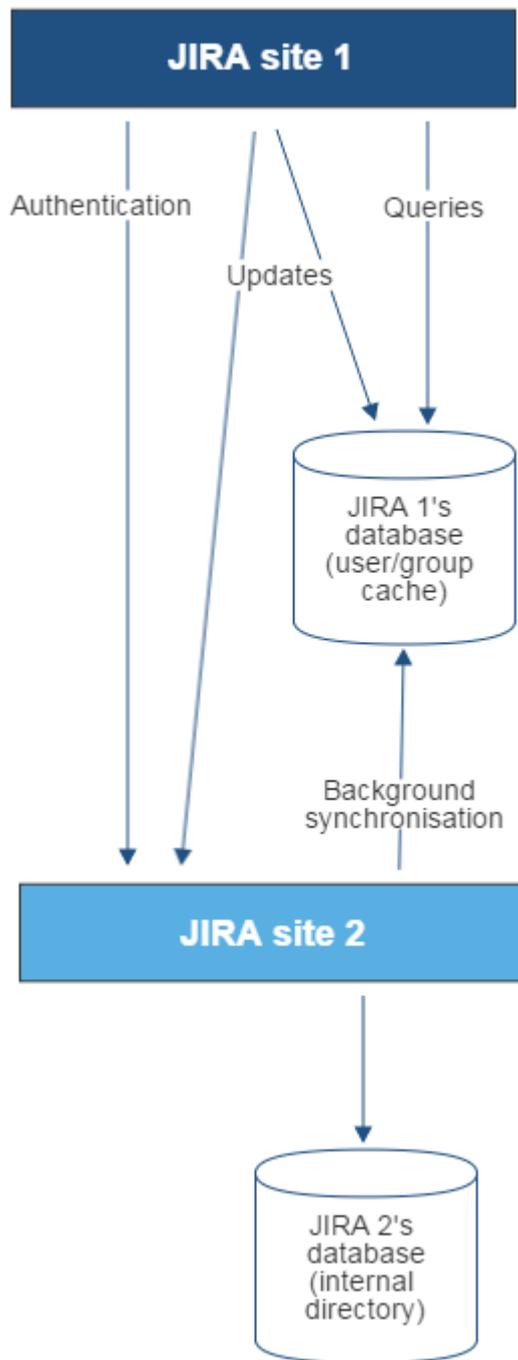


Diagram above: One JIRA site connecting to another for user management. JIRA site 2 does the user management, storing the user data in its internal directory.

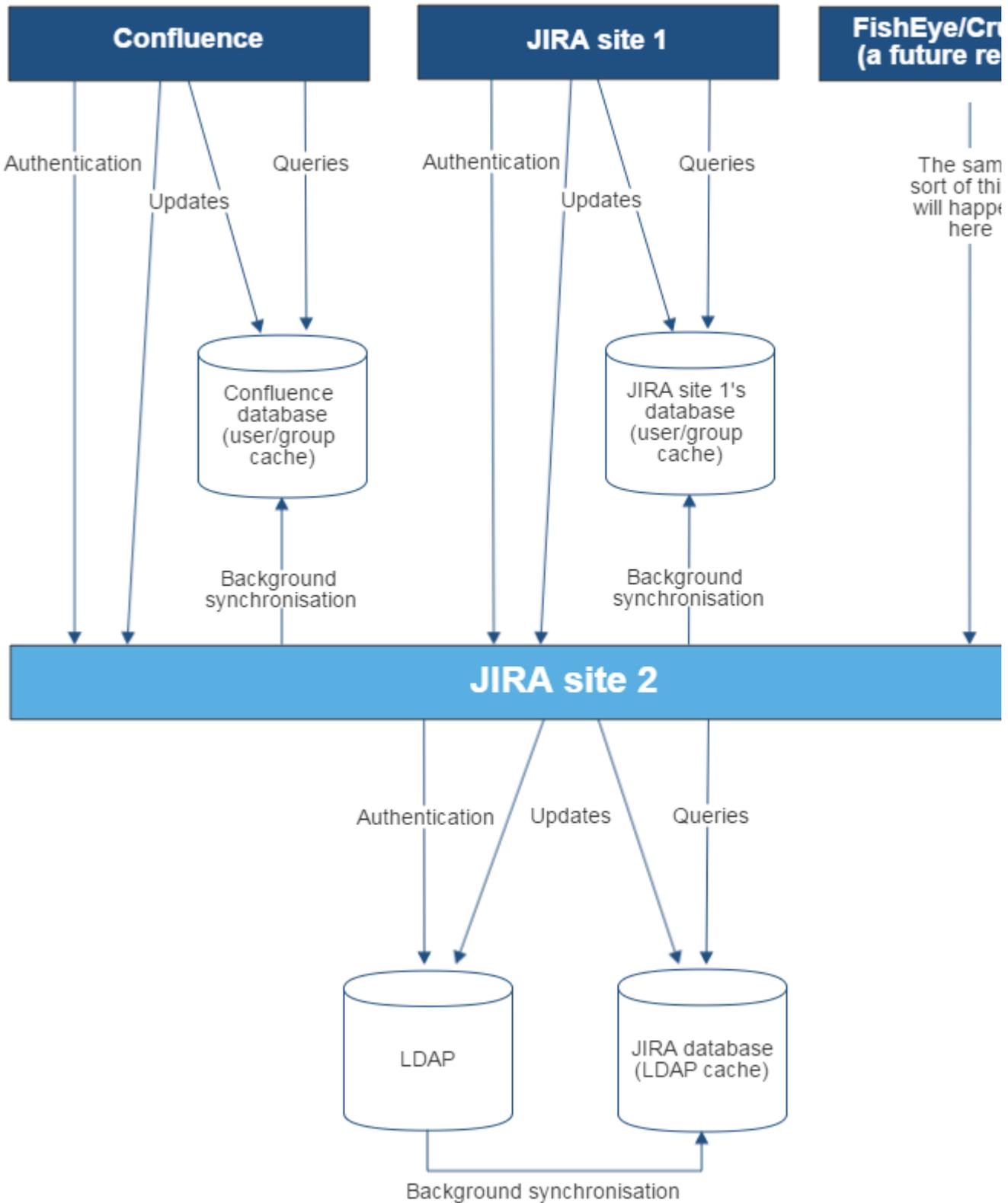


Diagram above: A number of applications connecting to JIRA (site 2) for user management, with JIRA in turn connecting to an LDAP server.

Related topics

### Configuring user directories

- [Configuring the internal directory](#)
- [Connecting to an LDAP directory](#)
- [Connecting to an internal directory with LDAP authentication](#)

- [Connecting to Crowd or another Jira application for user management](#)
- [Managing multiple directories](#)
- [Migrating users between user directories](#)
- [Synchronizing data from external directories](#)

# Managing multiple directories

This page describes what happens when you have defined more than one user directory in Jira. For example, you may have an internal directory and you may also connect to an LDAP directory server and/or other types of user directories. When you connect to a new directory server, you also need to define the **directory order**.

**Avoid duplicate usernames across directories.** If you are connecting to more than one user directory, we recommend that you ensure the usernames are unique to one directory. For example, we do not recommend that you have a user `jsmith` in both 'Directory1' and 'Directory2'. The reason is the potential for confusion, especially if you swap the order of the directories. Changing the directory order can change the user that a given username refers to.

## On this page:

- [Configuring the Directory Order](#)
- [Effect of Directory Order](#)
  - [Login](#)
  - [Permissions](#)
  - [Updating Users and groups](#)

Here is a summary of how the directory order affects the processing:

- The order of the directories is the order in which they will be searched for users and groups.
- Changes to users and groups will be made only in the first directory where the application has permission to make changes.

 Managing 500+ users across Atlassian products? Find out how easy, scalable and effective it can be with Crowd! See [centralized user management](#).

## Configuring the Directory Order

You can change the order of your directories as defined to Jira. Select '**User Directories**' from the Jira administration menu and click the blue up- and down-arrows next to each directory.

Directory Name	Type	Order
JIRA Internal Directory	Internal	 
LDAP server	OpenLDAP (Read-Write)	 

 In situations where users are unable to change their passwords, check that a Delegated Authentication Directory is not the highest in the order of User Directories. As a workaround, you can change the order of User Directories, or alternatively use a connection to a LDAP directory instead.

Notes:

- Please read the rest of this page to understand what effect the directory order will have on authentication (login) and permissions in Jira, and what happens when you update users and groups in Jira.

## Effect of Directory Order

This section summarizes the effect the order of the directories will have on login and permissions, and on the updating of users and groups.

### Login

The directory order is significant during the authentication of the user, in cases where the same user exists in multiple directories. When a user attempts to log in, the application will search the directories in the order specified, and will use the credentials (password) of the *first occurrence of the user* to validate the login attempt.

## Permissions

The directory order is significant when granting the user permissions based on group membership. If the same username exists in more than one directory, the application will look for group membership only in the first directory where the username appears, based on the directory order.

Example:

- You have connected two directories: The Customers directory and the Partners directory.
- The Customers directory is first in the directory order.
- A username `jsmith` exists in both the Customers directory and the Partners directory.
- The user `jsmith` is a member of group `G1` in the Customers directory and group `G2` in the Partners directory.
- The user `jsmith` will have permissions based on membership of `G1` only, not `G2`.

## Updating Users and groups

If you update a user or group via the application's administration screens, the update will be made in the first directory where the application has write permissions.

Example 1:

- You have connected two directories: The Customers directory and the Partners directory.
- The application has permission to update both directories.
- The Customers directory is first in the directory order.
- A username `jsmith` exists in both the Customers directory and the Partners directory.
- You update the email address of user `jsmith` via the application's administration screens.
- The email address will be updated in the Customers directory only, not the Partners directory.

Example 2:

- You have connected two directories: A read/write LDAP directory and the internal directory.
- The LDAP directory is first in the directory order.
- Since you can create users in both directories, you can choose the directory in which you want to perform the update.

Related topics

### Configuring user directories

- [Configuring the internal directory](#)
- [Connecting to an LDAP directory](#)
- [Connecting to an internal directory with LDAP authentication](#)
- [Connecting to Crowd or another Jira application for user management](#)
- [Managing multiple directories](#)
- [Migrating users between user directories](#)
- [Synchronizing data from external directories](#)

# Migrating users between user directories

Organizations will often migrate to or from LDAP engines, such as Active Directory or OpenLDAP, as they grow or acquire new companies, and need to migrate users into the same LDAP engine. As changes occur outside of Jira, they will also need to be reflected within the Jira user directories:

- Jira can have multiple user directories (e.g. Jira Internal, Delegated LDAP, LDAP Connector).
- The difference between the two is a connector will periodically synchronize user details against LDAP and can add/delete users and groups during that process. A delegated directory can only add users /groups upon the user's first login.
  - 📘 You can easily identify this by looking for the **Synchronize** option.
- Each directory will have **unique** users, groups and group memberships. This means there can be multiple users of the same username with different group memberships.
- Project Roles are global across all user directories.
- If you have the same user in multiple directories, the [effect of directory order](#) will apply. This means that if you add a new user directory and then change the order, so it is before your existing directory, your users will be selected from that directory first.
- When deactivating a user in LDAP, it will be deactivated in Jira.
- When deleting a user in LDAP, it will be deleted in Jira if it is not needed, or deactivated if it is (e.g. the user has comments).
- You can set up a User Directory with different [permissions settings](#) that will allow you to administer the groups in either LDAP, Jira, or both.

## On this page:

- [Using the 'migrate users from one directory to another' functionality](#)
- [Migrating users by changing the directory order](#)
- [Migrating users manually](#)

This guide describes how to migrate users between the different user directories, as described in [Configuring user directories](#). You will need to log in as a user with the Jira System Administrators' [global permission](#) to access the Settings menu.

📘 Managing 500+ users across Atlassian products? Find out how easy, scalable and effective it can be with Crowd! See [centralized user management](#).

## Using the 'migrate users from one directory to another' functionality

This functionality allows for the following scenarios:

- Migrate **all users** from Jira Internal to Delegated LDAP
- Migrate **all users** from Delegated LDAP to Jira Internal
- Migrate **all users** from Delegated LDAP to Delegated LDAP

However, it cannot be used for any of the following scenarios:

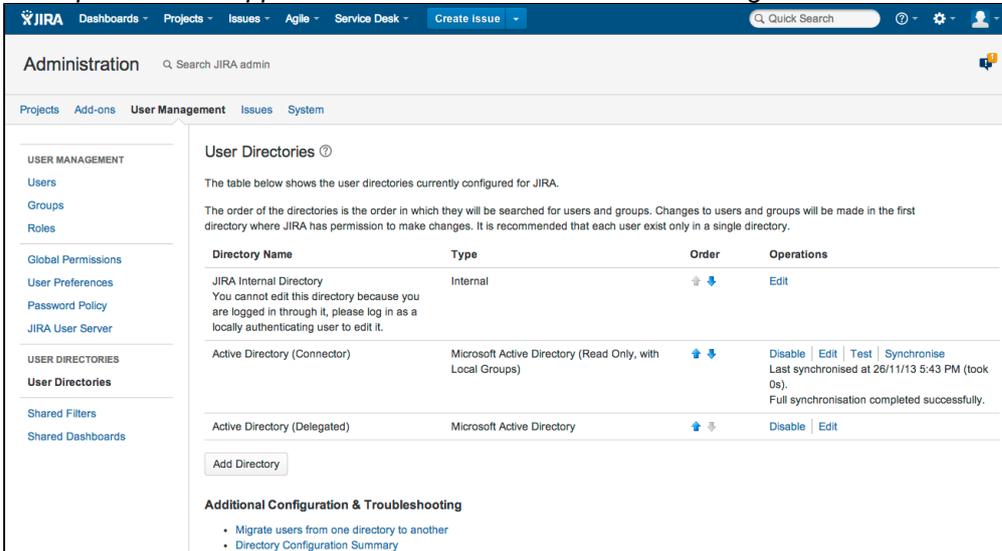
- Migrating a specific set of users or one single user from one directory to another
- Connector user directories these can be easily identified, as they have a Synchronize option
- Migrating groups only
- Migrating users without their groups

It also has the following features:

- If you, the currently logged-in user, are in the directory to be migrated from, your user data will **not** be migrated.
- Users and groups will not be migrated if they already exist in the target directory. For example, consider a user that exists in Jira Internal and Jira Delegated LDAP but has different groups in Jira Internal: when migrating from Jira Internal to the Jira Delegated LDAP, that user will be skipped and the groups will not be migrated.

## To migrate users:

1. If the username needs to be changed as part of the migration, rename them (see [Managing users](#) for instructions).
2. Choose **Administration**  > **User Management**.
3. Choose **User Directories**.
4. Choose **Additional Configuration & Troubleshooting** (section) > **Migrate users from one directory to another**.
5. *This option will not appear if there are no valid directories to migrate from/to.*

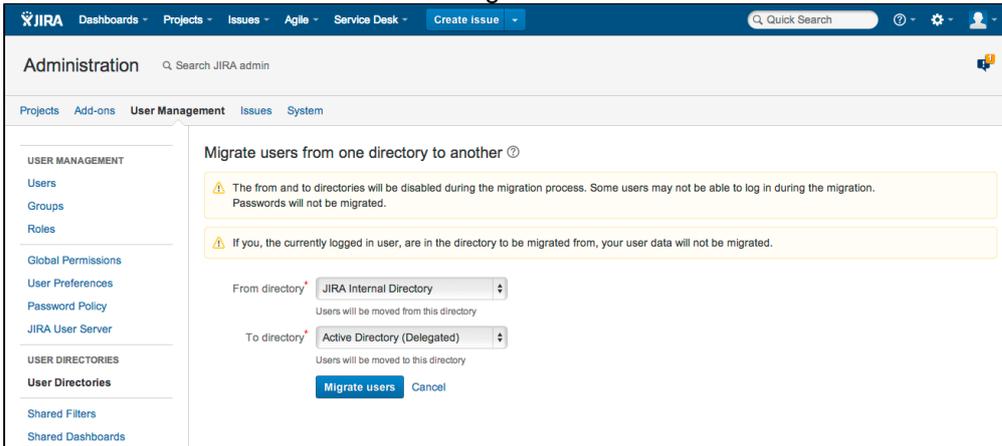


The screenshot shows the Jira Administration interface. The left sidebar contains navigation options: USER MANAGEMENT (Users, Groups, Roles), Global Permissions, User Preferences, Password Policy, JIRA User Server, USER DIRECTORIES (User Directories), Shared Filters, and Shared Dashboards. The main content area is titled "User Directories" and includes a table of configured directories:

Directory Name	Type	Order	Operations
JIRA Internal Directory	Internal	1	Edit
You cannot edit this directory because you are logged in through it, please log in as a locally authenticating user to edit it.			
Active Directory (Connector)	Microsoft Active Directory (Read Only, with Local Groups)	2	Disable   Edit   Test   Synchronise Last synchronised at 26/11/13 5:43 PM (took 0s). Full synchronisation completed successfully.
Active Directory (Delegated)	Microsoft Active Directory	3	Disable   Edit

Below the table is an "Add Directory" button and a section for "Additional Configuration & Troubleshooting" with links to "Migrate users from one directory to another" and "Directory Configuration Summary".

6. Select the from and to directories and migrate the users:



The screenshot shows the "Migrate users from one directory to another" configuration page. It includes two warning messages:

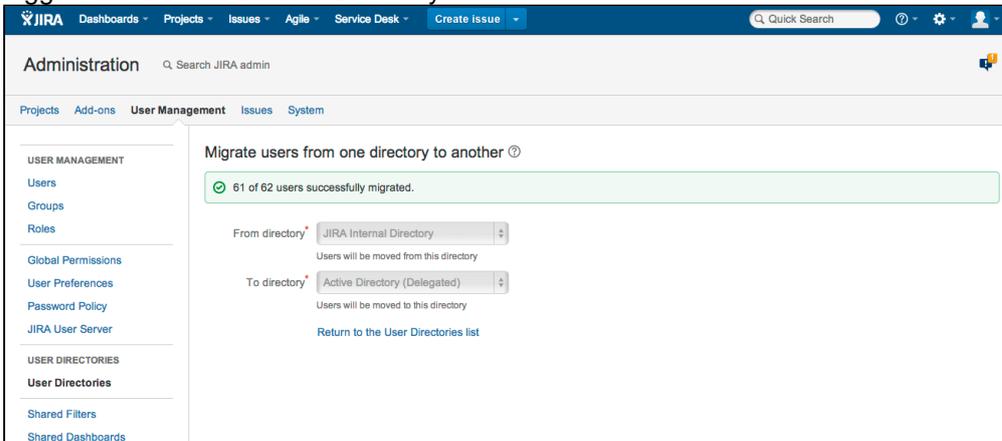
- The from and to directories will be disabled during the migration process. Some users may not be able to log in during the migration. Passwords will not be migrated.
- If you, the currently logged in user, are in the directory to be migrated from, your user data will not be migrated.

Below the warnings, there are two dropdown menus:

- From directory:** JIRA Internal Directory (Users will be moved from this directory)
- To directory:** Active Directory (Delegated) (Users will be moved to this directory)

At the bottom, there are "Migrate users" and "Cancel" buttons.

7. You will be shown a message telling you whether the migration was successful or not. In these example screenshots, only 61 out of 62 users could be migrated, as the user doing the migration was logged into the Jira Internal Directory.

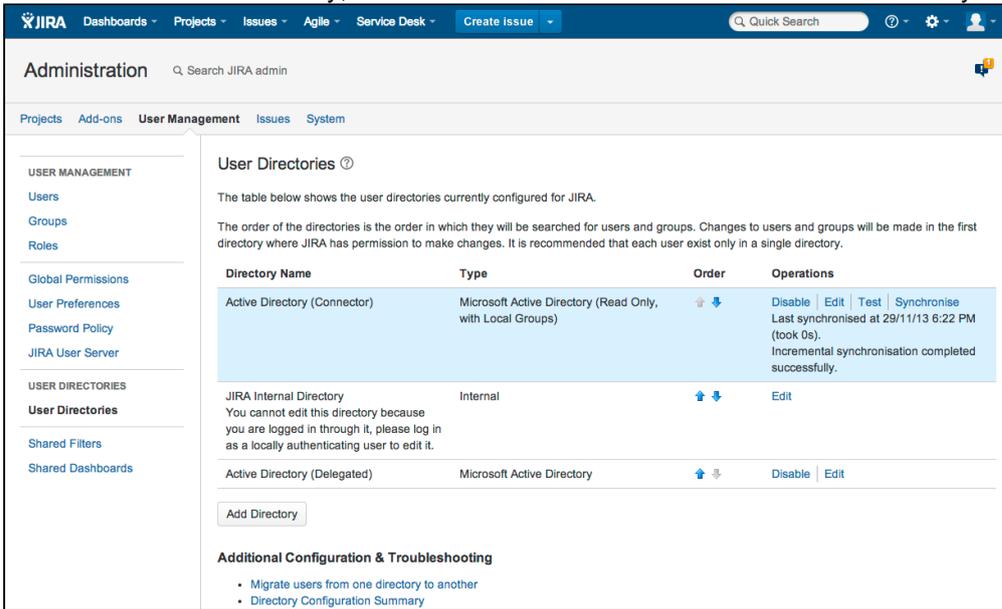


The screenshot shows the same "Migrate users from one directory to another" page, but with a success message at the top: "61 of 62 users successfully migrated." Below the message, the "From directory" and "To directory" dropdowns remain the same. A "Return to the User Directories list" link is now visible at the bottom.

Migrating users by changing the directory order

This method is only applicable if moving users from the Jira Internal Directory into an LDAP Connector and when LDAP will manage all their groups. Migrating users in this method will not move across any groups as the groups are separate from the Jira Internal Directory to the LDAP Connector.

1. Add the LDAP Connector, as detailed in [Connecting to an LDAP directory](#).
2. Move the new user directory, so that it is ordered before the Jira Internal Directory:



The screenshot shows the Jira Administration console with the 'User Directories' page. The page title is 'User Directories' and it includes a search bar for 'Search JIRA admin'. The left sidebar shows navigation options like 'Users', 'Groups', 'Roles', 'Global Permissions', 'User Preferences', 'Password Policy', 'JIRA User Server', 'User Directories', 'Shared Filters', and 'Shared Dashboards'. The main content area shows a table of user directories:

Directory Name	Type	Order	Operations
Active Directory (Connector)	Microsoft Active Directory (Read Only, with Local Groups)	1	Disable   Edit   Test   Synchronise Last synchronised at 29/11/13 6:22 PM (took 0s). Incremental synchronisation completed successfully.
JIRA Internal Directory	Internal	2	Edit
Active Directory (Delegated)	Microsoft Active Directory	3	Disable   Edit

Below the table, there is an 'Add Directory' button and a section for 'Additional Configuration & Troubleshooting' with links to 'Migrate users from one directory to another' and 'Directory Configuration Summary'.

When users login, they will login to the LDAP Connector rather than the Jira Internal Directory provided the usernames are identical.

## Migrating users manually

If the user migration does not fall into the above scenario, you can migrate users by modifying the database. See this knowledge base article for instructions on how to do this: [Move local group memberships between directories in Jira server](#). When

**JRASERVER-27868** - Provide the ability to migrate users from one directory to another

**FUTURE CONSIDERATION**

is completed, Jira will handle this in product.

# Synchronizing data from external directories

For certain directory types, Jira stores a cache of directory information (users and groups) in the application database, to ensure fast recurrent access to user and group data. A synchronization task runs periodically to update the internal cache with changes from the external directory.

## On this page:

- [Affected Directory Types](#)
- [How it Works](#)
- [Finding the Time Taken to Synchronize](#)
- [Manually Synchronizing the Cache](#)
- [Configuring the Synchronization Interval](#)

 Managing 500+ users across Atlassian products? Find out how easy, scalable and effective it can be with Crowd! See [centralized user management](#).

## Affected Directory Types

Data caching and synchronization apply to the following user directory types:

- **LDAP** (Microsoft Active Directory and all supported LDAP directories) where permissions are set to **read only**.
- **LDAP** (Microsoft Active Directory and all supported LDAP directories) where permissions are set to **read only, with local groups**.
- **LDAP** (Microsoft Active Directory and all supported LDAP directories) where permissions are set to **read/write**.
- **Atlassian Crowd**.
- **Atlassian JIRA**.

Data caching and synchronization do not occur for the following user directory types:

- **Internal Directory with LDAP Authentication**.
- **Internal Directory**.

## How it Works

Here is a summary of the caching functionality:

- The caches are held in the application database.
- When you connect a new external user directory to the application, a synchronization task will start running in the background to copy all the required users, groups and membership information from the external directory to the application database. This task may take a while to complete, depending on the size and complexity of your user base.
- Note that a user will not be able to log in until the synchronization task has copied that user's details into the cache.
- A periodic synchronization task will run to update the database with any changes made to the external directory. The default synchronization interval, or polling interval, is one hour (60 minutes). You can change the synchronization interval on the directory configuration screen.
  - Note for Confluence Data Center: The sync will take place on a single node of the cluster to update the database. This may make it seem like automatic synchronization will not be happening, but the task is assigned to one of the nodes.
- You can manually synchronize the cache if necessary.

- If the external directory permissions are set to read/write: Whenever an update is made to the users, groups or membership information via the application, the update will also be applied to the cache and the external directory immediately.
- All authentication happens via calls to the external directory. When caching information from an external directory, the application database does not store user passwords.
- All other queries run against the internal cache.

## Finding the Time Taken to Synchronize

The '**User Directories**' screen shows information about the last synchronization operation, including the length of time it took.

## Manually Synchronizing the Cache

You can manually synchronize the cache by clicking '**Synchronize**' on the '**User Directories**' screen. If a synchronization operation is already in progress, you cannot start another until the first has finished.

*Screen snippet: User directories, showing information about synchronization*

OpenLDAP	OpenLDAP (Read-Write)	↑ ↓	Disable Edit Synchronise Last synchronised at 14/01/11 3:07 PM (took 65s).
Crowd	Atlassian Crowd	↑ ↓	Disable Edit Synchronise Last synchronised at 14/01/11 2:39 PM (took 0s).

## Configuring the Synchronization Interval

You can set the '**Synchronization Interval**' on the directory configuration screen. The synchronization interval is the period of time to wait between requests for updates from the directory server.

The length you choose for your synchronization interval depends on:

- The length of time you can tolerate stale data.
- The amount of load you want to put on the application and the directory server.
- The size of your user base.

If you synchronize more frequently, then your data will be more up to date. The downside of synchronizing more frequently is that you may overload your server with requests.

If you are not sure what to do, we recommend that you start with an interval of 60 minutes (this is the default setting) and reduce the value incrementally. You will need to experiment with your setup.

Related topics

### Configuring user directories

- [Configuring the internal directory](#)
- [Connecting to an LDAP directory](#)
- [Connecting to an internal directory with LDAP authentication](#)
- [Connecting to Crowd or another Jira application for user management](#)
- [Managing multiple directories](#)
- [Migrating users between user directories](#)
- [Synchronizing data from external directories](#)

# Configuring projects

Jira projects are a way of grouping issues together and a way of applying the same sets of configurations to issues. These configurations, such as workflow, issue types and screens, can be changed on a per project basis, so that each project can have a different set of configurations. Setting up a Jira project effectively will enable your users to manage and complete their work quicker and more efficiently. This section of the documentation will take you through all the technical aspects of setting up your project, and give you information and tips on how to get the most out of your project.

Search the topics in 'Configuring projects':

Defining a project	<a href="#">Learn more</a> about creating, configuring and deleting a project. Find out what elements make up the configuration of a project, and how to change them.
Configuring issues	<a href="#">Learn more</a> about configuring your issue's fields, statuses, priorities and security, so that you can make your issues more effective for your organization.
Configuring permissions	<a href="#">Learn more</a> about configuring permissions, both specific to your individual project, and applicable to Jira as a whole.
Managing versions	<a href="#">Learn more</a> about versions, how to create, edit and delete a version, and how to use them to further group issues in your project.
Managing components	<a href="#">Learn more</a> about components, when and why to use them, and how to create, edit and delete them.
Screens, schemes and fields	<a href="#">Learn more</a> about how issue screens and schemes are set up and maintained, how to configure your issue's fields, and how to create notification schemes for your project.
Using the issue collector	<a href="#">Learn more</a> about how to configure and use the issue collector to get the most out of your projects.
Working with workflows	<a href="#">Learn more</a> about workflows and your project. Workflows define how your issues are managed in your project, and you can configure the workflow to perform specific actions when you work on your issues.

# Defining a project

This page tells you how to **add a new project**, **configure an existing project** or **convert an existing project to another project type**.

A Jira project is a collection of issues. Your team could use a Jira project to coordinate the development of a product, track a project, manage a help desk, and more, depending on your requirements. A Jira project can also be configured and customized to suit the needs of you and your team.

## On this page:

- [Before you begin](#)
- [Creating a project](#)
- [Convert a project type](#)
- [Re-index a project](#)
- [Archiving a project](#)
- [Delete a project](#)
- [Configuring a project](#)
- [A note about project administrators](#)

## Before you begin

For all of the following procedures, you must be logged in as a user with the [Jira Administrators global permission](#). A Jira administrator is able to create projects for all applications installed, but if the administrator does not have application access for that application, they will not be able to view the project after they have created it.

## Creating a project

1. Click **Projects** (in header) > **Create project**.
2. Follow the wizard to create the project.

About the project types:

- Depending on which Jira applications you have installed, you may have more than one project type available.
- Each project type has a specific set of features.
- All users on the Jira instance will be able to see all projects, but what features they see and what actions they can take are determined by their application access and the [project specific permissions](#).

About shared configurations:

- When you create a new project from a template, that project is created with its own set of schemes. These schemes are:
  - a permission scheme (default)
  - a notification scheme (default)
  - an issue security scheme
  - a workflow scheme
  - an issue type scheme
  - an issue type screen scheme
  - a field configuration scheme (default)
- Sometimes you may wish to share schemes among your projects, so that editing one scheme changes that scheme in several projects at once.
- You can select **Create with shared configuration** to select an existing project and to use that project's schemes. Note that when you're sharing schemes, any change to the scheme will affect all the projects using that scheme.

About the project details:

- The *project key* will be used as the prefix of this project's issue keys (e.g. 'TEST-100'). Choose one that is descriptive and easy to type.

- The *project lead* is a unique project role. Choose the person who manages the project as the project lead. If there is only one user in your Jira system, the Project Lead will default to that person and this field will not be available.
- If you're creating a project using a project type related to an application you currently do not have access to, Jira will display a checkbox that will allow you to grant yourself access to that application. This will add you to the default group of that application, and you will count as a user for that license.

## Convert a project type

At some point you may wish to convert an existing project to a different project type. For instance, you can convert a Jira Software project to a Jira Core project at the end of a Jira Software evaluation period, or when your team grows. You can only convert to project types of Jira applications that you have installed. Note that a project administrator may also change the project type.

1. Choose **Administration** (⚙️) > **Projects**, and select the relevant project..
2. Select **Details** in the **Project settings** menu.
3. Change the project type, and click **Save details**. Only project types for applications you have installed will be available.

You can review more on project types and what your users will see on the [project type and application overview](#) page.

## Re-index a project

To provide fast searching, Jira creates an index of the text entered into issue fields. It's sometimes necessary to regenerate this index manually; for instance if issues have been manually entered into the database, or the index has been lost or corrupted. You [regenerate the index for your entire Jira instance](#), or you can do it on a per project basis. Follow these instructions to re-index a single project.

1. Choose **Administration** (⚙️) > **Projects**, and select the relevant project..
2. Select **Re-index project** in the **Project settings** menu.
3. Click **Start project re-index**.

## Archiving a project

If your team has completed a project, you can archive it, so it doesn't stick around if it's no longer needed. An archived project will no longer appear in Jira, but you can keep viewing archived issues as read-only either through direct links or mentions in other projects.

1. Choose **Administration** (⚙️) > **Projects**.
2. Find a project you want to archive, and select **More** (⋮) > **Archive**. The project will be moved to the *Archived projects* page, which you can find in the left-hand navigation.

For more information about archiving projects, see [Archiving a project](#).

## Delete a project

If you're thinking about deleting a project from your Jira instance, remember that you can't reinstate it from within Jira. Once the project's been deleted, it can only be recovered by reinstating a backup or an XML copy, and this is no trivial task. Make sure you're comfortable deleting the project before proceeding. Deleting a project will only delete the issues, versions and components associated with the project. It won't delete any of the associated schemes, workflows or issues types, or any content that could be shared with another project.

1. Choose **Administration** (⚙️) > **Projects**, and select the relevant project..
2. Select **Delete project** in the **Project settings** menu.
3. Click **Delete** to begin deleting the project.
4. Acknowledge the project has been deleted.

## Configuring a project

1. Navigate to the settings page for the project by doing either of the following:
  - Choose **Administration** (⚙️) > **Projects**, and select the relevant project.
  - Navigate to the desired project's summary via the **Project** drop-down, and click the **Project settings** link.
2. Use the links on the left to navigate between the different project settings. Read the sections below for a description of each setting.

[Project details](#) | [Issue types](#) | [Workflows](#) | [Screens](#) | [Fields](#) | [Settings](#) | [Roles](#) | [Versions](#) | [Components](#) | [Permissions](#) | [Notifications](#) | [Development tools](#)

## Project details

Click **Details** in the **Project settings** sidebar, and edit the project details as desired. Once you've completed your edits, don't forget to click the **Save** button. Note the following:

- Editing the project key: This is not a simple task. Read this page before you edit the project key: [Editing a project key](#).
- Using the Wiki Style Renderer in the project description: You can use the [Wiki Style Renderer](#) to display rich text (HTML) in your project description.
- Choosing a project avatar: If you don't want to use a project avatar, you can upload a transparent pixel. This effectively loads the transparent pixel, which means you won't see an image.

### About project categories:

Categories can be viewed/created via **Administration** (⚙️) > **Projects** > **Project Categories**.

Why are categories useful? Jira can search for all the issues in a particular project category (e.g. `category = "buildeng"` in an advanced search), and can display projects sorted by the project category. A Jira project can only belong to one category. Please note that a project category is not part of a project hierarchy. Also, Jira does not support sub-projects or parent projects.

## Issue types

Jira enables you to keep track of different types of things – bugs, tasks, helpdesk tickets, etc – by using different *issue types*. You can also configure each issue type to act differently, e.g. to follow a different process flow or track different pieces of information.

Click either **Issue Types** in the left menu or one of the issue types under it, e.g. **Bug**, **Task**, **Story**, etc:

- **Issue Types**: Click this to configure which issue types apply to this project (choose an [issue type scheme](#) or [edit the existing scheme](#)). You can also configure the workflow, fields and screens for the issue type in the project, but it is easier to do this by clicking one of the issue types.
- **One of the issue types (e.g. Bug, Task, Story)**: Click this to configure the workflow/screen for the issue type in the project. The workflow screen (**Workflow** tab) shows the [workflow designer](#). The screen (**View** tab) shows the [screen designer](#).

## Workflows

Your Jira issues can follow a process that mirrors your team's practices. A *workflow* defines the sequence of steps (or statuses) that an issue will follow, e.g. Open, In Progress, Resolved. You can configure how issues will transition between statuses, e.g. who can transition them, under what conditions, and which screen will be displayed for each transition.

- **Workflow Scheme** the project's [workflow scheme](#) determines which [workflows](#) (issue state transitions) apply to issue types in this project.

## Screens

Jira allows you to display particular pieces of issue information at particular times, by defining *screens*. A screen is simply a collection of fields. You can choose which screen to display when an issue is being created, viewed, edited, or transitioned through a particular step in a workflow.

- **Screen Scheme** the project's [screen scheme](#) determines which [screens](#) are displayed for different issue operations (view, edit, create);  
**OR**
- **Issue Type Screen Scheme** the project's [issue type screen scheme](#) determines which [screens](#) are displayed for different issue operations (view, edit, create), for different issue types.

## Fields

Jira enables you to define field behavior: each field can be required/optional, rich text/plain text, hidden/visible. You define this behavior by using a *field configuration*.

- **Field Configuration Scheme** the project's [field configuration scheme](#) determines which [field configuration](#) applies to issue types in this project. (A [field configuration](#) determines each field's overall visibility, requiredness, formatting (wiki/rich-text or plain) and help-text).

## Settings

- **Application Links** (Configure project links) if you have linked your Jira instance to other Atlassian applications, like Confluence, FishEye or other Jira instances, you will be able to link this Jira project to areas of those applications that contain information relating to your project or team. For example, Confluence spaces, FishEye repositories, Jira projects (in another Jira instance), etc. This allows you to take advantage of integration points between these applications. See [Using AppLinks to link to other applications](#) for information about application links and project links.

## Roles

Different people may play different roles in different projects the same person may be a leader of one project but an observer of another project. Jira enables you to allocate particular people to specific roles in your project.

- **Project Lead** user fulfilling the role of project leader. Used as the 'Default Assignee' (except for Jira Software projects where it is set to 'Unassigned'), and potentially elsewhere in Jira (e.g. in permission schemes, notification schemes, issue security schemes and workflows).
- **Default Assignee** the user to whom issues in this project are initially assigned when created. Can be either the 'Project Lead' (above), or, if **Allow unassigned issues** is set to 'On' in Jira's [general configuration](#), 'Unassigned'. There are also [default component assignees](#).  
  - By default, new projects also have their 'Default Assignee' set to 'Unassigned.' You can change this here if you want to set it to be a specific role, i.e. 'Project Lead.'
- **Project Roles** members are users/groups who fulfil particular functions for this project. [Project roles](#) are used in permission schemes, notification schemes, issue security schemes and workflows.

## Versions

Issues can be grouped in Jira by allocating them to versions. For example, if you are using Jira to manage the development of a product or manage the build of a house, you may want to define different *versions* to help you track which issues relate to different phases of your product or build (e.g. 1.0, 1.1, 1.2, 2.0, 2.0.1). Jira can help you manage, release and archive your versions. Versions can also have a Release Date, and will automatically be highlighted as "overdue" if the version is unreleased when this date passes.

- **Versions** versions defined in the project. See the [version management](#) page for details.

## Components

You may want to define various *components* to categorize and manage different issues. For a software development project, for example, you might define components called "Database", "Usability", "Documentation" (note that issues can belong to more than one component). You can choose a Default Assignee for each component, which is useful if you have different people leading different sub-teams in your project.

- **Components** logical groups that this project's issues can belong to. See the [component management](#) page for details.

## Permissions

Jira allows you to control who can access your project, and exactly what they can do (e.g. "Work on Issues", "Comment on Issues", "Assign Issues"), by using *project permissions*. You can also control access to individual issues by using *security levels*. You can choose to grant access to specific users, or groups, or roles (note that roles are often the easiest to manage).

- **Permission Scheme** the project's [permission scheme](#) determines who has permission to view or change issues in this project.
- **Issue Security Scheme** the project's [issue security scheme](#) determines what visibility levels issues in this project can have.

## Notifications

Jira can notify the appropriate people when a particular event occurs in your project (e.g. "Issue Created", "Issue Resolved"). You can choose specific people, or groups, or roles to receive *email notifications* when different events occur. (Note that roles are often the easiest to manage.)

- **Notification Scheme** the project's [notification scheme](#) determines who receives email notifications of changes to issues in this project.
- **Email** specifies the 'From' address for emails sent from this project. Only available if an SMTP email server has been configured in Jira.

⚠ Please note, the **Default Notification Scheme** (shipped with Jira) is associated with all new projects by default. This means that if you have an outgoing (SMTP) mail server set up, that email notifications will be sent as soon as there is any activity (e.g. issues created) in the new project.

## Development tools

The Development tools section is only available on Jira Software projects, and can only be viewed by Jira Software users. It gives you an overview of the development tools that are connected and which users can use the integration features between them:

- **View permission** - This section lists which users can see the development tools integration features (like the **Create Branch** link) on the view issue screen, as well as other development-related information, like commits, reviews and build information. This ability is controlled by the "View Development Tools" project permission.
- **Applications** - This section shows which development tools are connected to Jira via application links and are eligible to use the development tool features in Jira.

## A note about project administrators

A project administrator in Jira is someone who has the project-specific **Administer Projects** [project permission](#), but not necessarily the **Jira Administrator** [global permission](#).

Without the **Jira Administrator** [global permission](#), however, project administrators can do the following:

- Edit the project name
- Edit the project description
- Edit the project avatar image
- Edit the project URL
- Edit the project lead
- Edit [project role membership](#)
- Change the project type
- Define [project components](#)
- Define [project versions](#)
- View, but not select nor edit the project's schemes (notification scheme, permission scheme, etc)

Changing the project category of a Jira project requires **Jira Administrator** [global permission](#).

# Editing a project key

Editing a project key is not a trivial task. You should choose key that will suit your long-term needs when creating a project, rather than rely on editing the project key after the project is created. However, there are situations where you need to change the key for an existing project, e.g. change of product name.

The instructions on this page show you how to change the project key and describe the implications of such a change.

- [Before you begin](#)
- [Editing the project key](#)
- [Notes for change management](#)
- [Related topics](#)
- [Notes for developers](#)

## Before you begin

- Your desired project key must conform to the project key format restrictions specified in your Jira applications. By default, the project key format must be at least 2 characters long and contain only uppercase letters. You can change the project key format to enforce different restrictions. See [Changing the project key format](#) for instructions.
- Perform this change during a low usage period Jira applications will start a [background re-index](#) when you save your updated project key. This can have a performance impact on your instance. Note, you cannot choose a 'Lock Jira and rebuild index'. The background index will be faster anyway, as it is limited to issues for the project.
- Communicate changes to your users Ensure that you are aware of the consequences of changing the project key, and have adequately prepared your users for the changes. See the [Changes](#) section below.

## Editing the project key

1. Choose **Administration**  > **Projects**, and select the relevant project..
2. Locate the project that you'd like to change.
3. Select **Edit** in the **Actions** column of the project you want to change.
4. Edit the project key, and click **Save details**. Only project types for applications you have installed will be available.

This will start a project re-index, which you need to acknowledge when it finishes.

### Note:

- If you update any other project detail fields, you'll see the changes immediately. You won't need to wait for the re-index to finish.
- If you cancel the background re-index, you will have trouble searching for issues related to the project. If you do need to cancel it, you can run it again later to fix these problems.

## Post-update tasks

- **Fix the project entity links** When you connected Jira to another Atlassian application, entity links would have been automatically created between your Jira projects and the relevant "projects" in other applications, e.g. Confluence spaces. If you change the key of a Jira project, you will need to fix the project entity links, as described in [Creating links between projects](#).
- **Updating Jira Software agile board filters**- If your Jira Software agile boards use the old project key, the board filters may need to be updated to reflect the new project key. Otherwise the board might not display issues from the renamed project.

## Notes for change management

While editing the project key is a major change, in most cases, your Jira project will work as you'd expect with a new key. There are a few cases that you should be aware of, which are listed below. We recommend reviewing these and advising your users accordingly.

- The old project key can be used in JQL queries. Users won't have to update issue filters that reference the old project key.
- If you use Confluence with Jira, the Jira issue macros in Confluence will continue to work. Please note, if you don't see the change straight away, allow some time for the cache to refresh.
- You won't be able to create a new project with the old project key. However, you can change the renamed project back to the old project key. If you delete the project, all associated keys will be freed and you'll be able to re-use them.
- Links will work, whether they are inside Jira or from external sources. However, link aliases will not be updated. For example, if you have a link to an issue 'EXAMPLE-1' in the description of an issue, and you change the project key 'EXAMPLE' to 'DEMO', then the alias 'EXAMPLE-1' will not be updated to 'DEMO-1'. The link will still direct you to DEMO-1 though.
- If you are using a gadget with a global filter, you will need to update the filter after the project is renamed.
- All attachments will be accessible after the project key change. Please note, however, that the directory that they are stored in (under the `<Jira Home>\data\attachments\directory`) will retain the old project key. For example, if you change a project's key from TEST to DEMO, the attachments will be stored under `<Jira Home>\data\attachments\TEST`.
- If you export a renamed project, and then import it, it will have the updated project key, i.e. the original project key will not be retained. In fact, all historical keys for that project will be removed. There is a workaround for this that involves changing data directly in your database, see this [Answers post](#).

## Related topics

**Changing the maximum project key length** You can change the maximum characters allowed for a project key. Navigate to the General Configuration page of the Jira administration console, as described on [Configuring Jira application options](#), and change the **Maximum project key size** field. **Changing the project key format** You can change the format of a project key. This restricts the format of a project key when it is [created](#) or edited (as described above). For instructions, see [Changing the project key format](#).

## Notes for developers

- REST API calls will still work with old project key. REST calls that specify an issue key will work with the old issue key after the project key has changed. For example, `/rest/api/issue/EXAMPLE-100` will still work after the project key is changed from EXAMPLE to DEMO.
- We have created a new event, `ProjectUpdatedEvent`. This event is triggered any time a project's details are changed, including changing the project key.
- If you need to retrieve all issue keys and project keys (historical and current), you can do this via the following:
  - REST:
    - Get all project keys for a project: `/rest/api/2/project/<project key>?expand=projectKeys`
  - Java API:
    - Get all project keys: `com.atlassian.jira.project.ProjectManager#getAllProjectKeys`
    - Get all issue keys for an issue: `com.atlassian.jira.issue.IssueManager#getAllIssueKeys`

# Changing the project key format

Jira provides the ability to specify the format of project keys within the system. This allows you to restrict the format of a project key, when a project key is [created](#) or [edited](#).

A project key format is defined via a regular expression 'rule' that governs the valid project key format. By default, the Jira project key configuration requires two or more uppercase alphabetical characters based on the regular expression `([A-Z][A-Z]+)`.

## On this page:

- [Before you begin](#)
- [Configuring the project key format](#)
- [Related topics](#)

## Before you begin

- Ensure that you choose a **supported project key format**. Only formats that meet all of the following rules are supported:
  - The first character must be a letter,
  - All letters used in the project key must be from the [Modern Roman Alphabet](#) and upper case, and
  - Only letters, numbers or the underscore character can be used.Examples:
  - Examples of supported keys: `PRODUCT_2013,R2D2,MY_EXAMPLE_PROJECT`.
  - Examples of unsupported keys: `2013PROJECT` (*first character is not a letter*), `PRODUCT-2012` (*hyphens are not supported*).
- You cannot configure the issue key pattern, as Jira expects this key to conform to specific rules. By default, Jira issue keys (or issue IDs) are of the format `<project key>-<issue number>`, e.g. `ABC-123`. For example, you can't show the issue number before the project key.
- If a number of issues have already been created in your Jira installation, then *changing the project key format is not recommended*. If you must change the project key pattern after issues have already been created, use a regular expression that allows a more 'permissive' project key pattern than the current one (e.g. use a regular expression which will still be valid for existing project keys defined in your Jira installation).

 If you have integrated Jira with [Bamboo](#), do not change Jira's default project key format as Bamboo only supports this key format.

## Configuring the project key format

The `jira.projectkey.pattern` property allows Jira administrators to specify a Perl5 regular expression value that defines the rule for a valid project key. Further information on Perl5 is available [here](#).

This property and its regular expression value can be defined through the **Advanced Settings** page. This is described below.

### Step 1. Configure a pattern for your project key syntax

1. Navigate to the Jira Advanced settings page, as described on [Configuring advanced settings](#).
2. Find the `jira.projectkey.pattern` property and click its value to modify it. Below is a list of common examples and patterns:

Pattern Requested	Expression needed	Resulting Issue IDs	Comments
XXYY, where X indicates two fixed letters, Y represents two fixed digits	<code>([A-Z]{2}[0-9]{2})</code>	TQ09-01, TQ09-02, etc.	<code>[A-Z]</code> Any character from A to Z <code>{2}</code> Matches the preceding character 2 times exactly <code>[0-9]</code> Any character (i.e. digit) from 0 to 9

XZ+, where X indicates one fixed letter, Z+ represents one or more letters, digits or underscore characters	([A-Z][A-Z_0-9]+)	ACAT_51-1, AAA5-1330, A_20_A091-15, etc.	[A-Z] Any characters from A to Z [A-Z_0-9] Any character from A to Z, 0 to 9 or the underscore character. + specifies [A-Z_0-9] as one or more characters from A to Z, 0 to 9 or the underscore character.
---	-------------------	--	--

Please note:

- Jira prepends the regular expression specified with '^' and closes it with '\$' for an exact matching rule within the system.
- The project key only supports uppercase characters, as stated above. Hence, for simplicity, use uppercase characters in your expressions as Jira will convert any lowercase characters to uppercase ones.

## Step 2. Test your regular expression

A variety of tools allow searching using a Regular Expression. Most text editors will allow a Regular Expression search. There are also a variety of websites available to for testing a Regular Expression available from an Internet search.

## (Optional) Step 3. Customize the project key description and warning

In addition to the project key format, you can also customize the following properties in the [jira-config.properties](#) file:

- `jira.projectkey.description` a configurable description (to match the project key pattern) displayed on project creation
- `jira.projectkey.warning` if Jira detects that the project key entered does not match the `jira.projectkey.pattern`, it will throw the error message defined in `jira.projectkey.warning`. You can change this error message, so that when a user keys in the wrong format, they will be informed of the correct pattern to use.

## Related topics

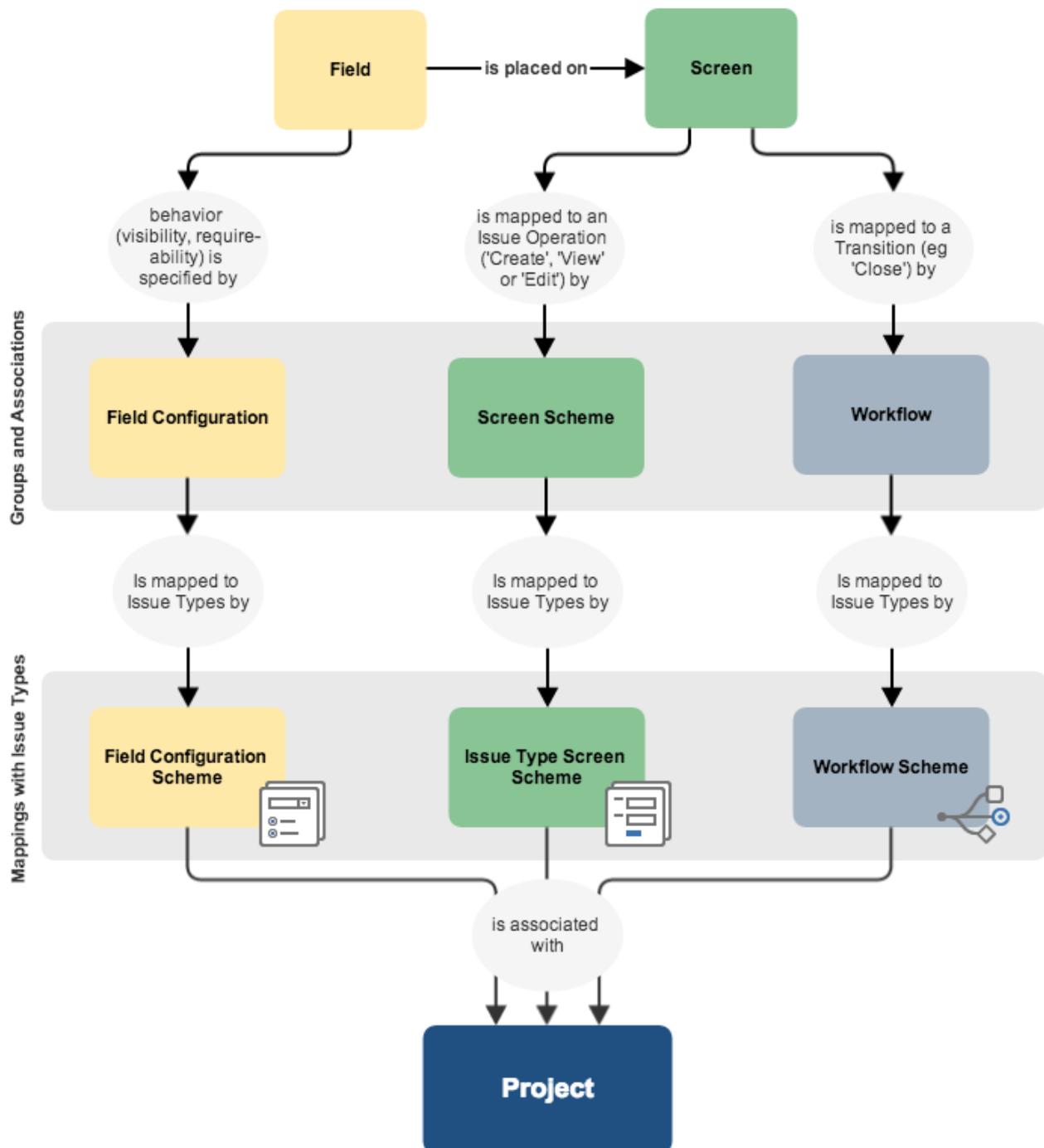
- **Changing the maximum project key length** You can change the maximum characters allowed for a project key. Navigate to the General Configuration page of the Jira administration console, as described on [Configuring Jira application options](#), and change the **Maximum project key size** field.
- [Defining a project](#)
- [Editing a project key](#)
- [Configuring advanced settings](#)

# Configuring issues

## Overview

To help you tailor Jira to your organization's needs, Jira enables you to manipulate the display and behavior of issue fields ('Summary', 'Description', 'Issue Type', etc). You can:

- [Change a field's description](#)
- [Make a field hidden or visible](#)
- [Make a field required or optional](#)
- Add your own values for [issue type](#), [priority](#), [resolution](#), and [status](#)
- Create new [custom fields](#)
- Enable a [rich text renderer](#) for (some) fields
- Position fields on a [screen](#)
- Choose which screen should be displayed for each [issue operation](#) (e.g. 'Create Issue', 'Edit Issue') or [workflow transition](#) (e.g. 'Resolve Issue', 'Close Issue')



## Concepts

Some key Jira concepts include:

- **Field configuration** a set of definitions for all fields, comprising: each field's description; whether each field is hidden or visible; whether each field is required or optional; and what type of renderer to use for each text field.
- **Screen** defines which fields are present on a screen, and their order. (Note that a hidden field can be present on a screen, but will still be invisible.)
- **Screen scheme** associates different screens with different issue operations (e.g. 'Create Issue', 'Edit Issue', 'View Issue').
- **Workflow** defines the steps (i.e. statuses) and transitions to other steps that an issue moves through during its lifecycle. Screens can also be mapped to different transitions of a workflow.
- **Field configuration scheme** associates field configurations with **issue types**, which in turn is applied to projects. This allows you to specify different behaviors for a field, for each type of issue in a given project.

- [Issue type screen scheme](#) associates screen schemes with [issue types](#), which in turn is applied to projects. This allows you to specify different screens for a particular operation (e.g. 'Create Issue'), for each type of issue in a given project. For example, you could use one screen when *creating an issue* of type 'Bug', and a different screen when *creating an issue* of type 'Task'.
- [Workflow scheme](#) associates Workflows with [issue types](#), which in turn is applied to projects. This allows you to specify different workflows for each type of issue in a given project.
- [Issue type scheme](#) is applied to projects and defines (or *restricts*) which [issue types](#) are available to those projects.
  - **i** If the [field configuration scheme](#), [issue type screen scheme](#), and [workflow scheme](#) associated with a given project contain associations with other issue types that are not specified in the project's [issue type scheme](#), then those other issue types will be ignored by the project since the project's Issue Type Scheme restricts what issue types the project can use.

## Related topics

- [Configuring built-in fields](#)
  - [Defining issue type field values](#)
    - [Associating issue types with projects](#)
  - [Defining priority field values](#)
    - [Associating priorities with projects](#)
  - [Defining resolution field values](#)
  - [Defining status field values](#)
  - [Translating resolutions, priorities, statuses, and issue types](#)
- [Issue fields and statuses](#)
- [Configuring issue-level security](#)

# Configuring built-in fields

Each issue has a number of built-in fields, and some of the built-in fields can be customized as follows:

- [Defining issue type field values](#)
  - [Associating issue types with projects](#)
- [Defining priority field values](#)
  - [Associating priorities with projects](#)
- [Defining resolution field values](#)
- [Defining status field values](#)
- [Translating resolutions, priorities, statuses, and issue types](#)

# Defining issue type field values

Jira applications ship with a set of default issue types to help you get started. You can add, edit and delete your own custom issue types to suit the needs of your team. The diagram on [Configuring issues](#) shows how issue types relate to other entities in Jira applications.

## On this page:

- [Creating an issue type](#)
- [Deleting an issue type](#)
- [Editing an issue type](#)

Note that you can also:

- Control the set of available issue types for each project see [Associating issue types with projects](#).
- Control the display order of available issue types and the default issue type for each project see [Associating issue types with projects](#).  
• [Reordering](#) issue types changes the order in which they are displayed to the user who is creating an issue; and the *default* issue type is the one that is displayed in the selection-box.
- Associate particular issue types with specific fields, screens and workflow for details see [Associating field behavior with issue types](#), [Associating screen and issue operation mappings with an issue type](#), and [Managing your workflows](#), respectively.

*Tip: You can quickly configure the workflow/screen design of an existing issue type for a project via the project administration page. See [Defining a project](#) for details.*

## Creating an issue type

When creating a new issue type in Jira applications, you can create either a new standard or sub-task issue type. However, to create a sub-task issue type, you must [enable sub-tasks](#).

You can also create sub-tasks on the **Sub-Tasks** page. See [Configuring sub-tasks](#) for details.

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Issue Types** to view all issue types used by your Jira applications.
3. Select **Add Issue Type** and enter the following details:
  - **Name** enter a short phrase that best describes your new issue type
  - **Description** enter a sentence or two to describe when this issue type should be used
  - **Type** specify whether the issue type you are creating is a **Standard** issue type or a **Sub-Task** issue type. Sub-tasks are associated with individual **Standard** issues. Note that this option will not be available if [sub-tasks are disabled](#).
4. Select **Add** to create your new issue type.  
• Your new issue type will be automatically added to the **Default Issue Type Scheme**. You may want to also add it to other issue type schemes for more information, see [Associating issue types with projects](#).

## Deleting an issue type

### Before you begin:

- If any issues of the Issue Type you are about to delete exist in your Jira installation, please ensure this Issue Type has the following requirements (to ensure Jira prompts you to choose a new Issue Type for those issues):
  - the same [workflow](#) in all [workflow schemes](#) that are associated with one or more projects.
  - the same [field configuration](#) in all [field configuration schemes](#) that are associated with one or more projects.
  - the same [screen scheme](#) in all [issue type screen schemes](#) that are associated with one or more projects.
- **Alternatively**, you can simply search for all issues that currently use the Issue Type which you are about to delete and perform a bulk move to change those issues to a different Issue Type.

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Issue Types** to open the Issue Types page, which lists all issue types.

3. Click the **Delete** link (in the **Operations** column) for the issue type that you wish to delete.
4. Complete the fields.

## Editing an issue type

1. Choose **Administration**  > **Issues**.
2. Select **Issue Types** to open the Issue Types page, which lists all issue types.
3. Click the **Edit** link (in the **Operations** column) for the issue type that you wish to edit.
4. Edit the **Name**, **Description** and/or **Icon** as described above for creating an issue type.

**Please note:** To reorder an Issue Type, or set it as a default, see [Associating issue types with projects](#). (*Reordering* issue types changes the order in which they are displayed to the user who is creating an issue; and the *default* issue type is the one that is displayed in the selection-box.)

# Associating issue types with projects

What is an 'issue type scheme'?

An 'issue type scheme' defines a subset of [issue types](#), which:

- restricts the set of available [issue types](#) for a project, and
- controls the order of available issue types and the default issue type shown to your users for a project.
  - 📌 The 'default issue type' is the issue type displayed in the selection-box when a user creates an issue.

A single issue type scheme can be 're-used' across multiple projects, so that a group of similar projects (i.e. projects which might be used for similar purposes) can share the same issue type settings.

For example, all projects in your company may fit one of two 'purpose' categories:

- Development-related projects or
- Support-related projects.

Hence, you could create one scheme called *Development Issue Type Scheme* (with issue types *Bug* and *Feature*) and another called *Support Issue Type Scheme* (with issue types *Development Query* and *Support Request*). You can then associate each of these schemes with the appropriate project(s), for which there may be a plethora.

This provides your users with a different set of issue types based on the project they decide to create issues in and furthermore reflects the purpose behind creating these issues.

Your future maintenance workload is minimized, because any change you make to an issue type scheme is made across all projects that are associated with the scheme. In the example above, adding a new issue type to all support-related projects only requires the simple step of adding the issue type to the *Support Issue Type Scheme*.

**⚠️ Note:** For all of the following procedures, you must be logged in as a user with the [Jira Administrators global permission](#).

## Managing issue type schemes

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Issue Types** > **Issue Type Schemes** to open the Issue Type Schemes page, which displays all existing issue type schemes, their related issue types and their associated projects.

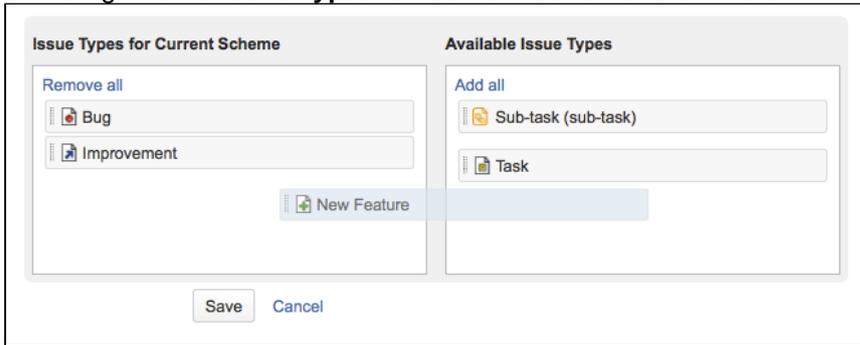
## Creating a new issue type scheme

1. Go to the **Issue Type Schemes** tab (see [above](#)).
2. Select **Add Issue Type Scheme** and enter a Scheme Name and Description.
  - 📌 Ensure that the Scheme Name is meaningful to other administrators, who will be able to reuse the scheme.

### On this page:

- [What is an 'issue type scheme'?](#)
- [Managing issue type schemes](#)
- [Choosing a project's issue type scheme](#)
- [Using the Issue Type Migration Wizard](#)

- To add issue types to your scheme, drag and drop an issue type from the **Available Issue Types** list on the right to the **Issue Types for Current Scheme** list on the left:



- If you need an issue type that does not currently exist, you can easily add this by using the **Add New Issue Type** button and dialog box. This will [add the issue type](#) to your Jira system and also add it to the **Issue Types for Current Scheme** list on the left.
- To reorder the issue types, drag and drop them into the preferred positions. *Reordering* issue types changes the order in which they are displayed in the selection-box when a user creates an issue.
- Set the **Default Issue Type** for the new scheme from the drop-down list. **Please Note:**
  - The 'default issue type' is the issue type displayed in the selection-box when a user creates an issue.
  - The issue types in this list depend on the issues in the **Issue Types for Current Scheme** list on the left.
  - The **None** option means that there is no default value. If this option is selected, the system will show the first Issue Type listed in the **Issue Types for Current Scheme**.
  - The **Issue Type** is remembered as long as you keep creating issues in the same project. Once you change projects or log off the system, it goes back to the *default* value.
- Click the **Save** button to create your issue type scheme.

### Editing an issue type scheme

- Go to the **Issue Type Schemes** tab (see [above](#)).
- Click the **Edit** link (in the **Operations** column) to access and edit the relevant issue type scheme.

#### **Please note:**

- The process of editing a scheme is identical to the creation process. While editing your issue type scheme, you can set the default default issue type and reorder, add or remove issue types.
- If an issue type scheme has been associated with one or more Jira projects ([below](#)) and:
  - issues of the issue types (defined by this issue type scheme) already exist in any of these Jira projects and
  - you then want to remove one or more of these issue types from this issue type scheme, you will be prompted to use the [Issue Type Migration Wizard](#) (below). This wizard will move your issues from the original issue type (which will no longer be applicable) to a valid one. If you cancel this process at any time, your changes will not be saved.

### Associating an issue type scheme with projects

- Go to the **Issue Type Schemes** tab (see [above](#)).
- Click the **Associate** link (in the **Operations** column) for the relevant Issue Type scheme.
- Using the multi-select **Project** box, choose the Jira projects that you wish to apply your issue type scheme to.
- Select **Associate** and all selected projects will change from their current scheme to the selected scheme.

**Please note:** If a project you are attempting to associate your new issue type scheme with has issues with issue types which have not been added to this new issue type scheme, you will be asked to use the [Issue Type Migration Wizard](#) (below) to migrate the issues to a new issue type (made available by the new issue type scheme).

### Choosing a project's issue type scheme

You may want to change a project to use a different set of issue types.

📌 This is effectively the same as associating an issue type scheme with projects ([above](#)), but is performed from a project's **Project Summary** administration page (and you cannot choose multiple projects in one action).

1. Choose **Administration**  > **Projects**, and select the relevant project.
2. In the **Issue Types** section, click the name of the current scheme to display the details of the project's issue type scheme.
3. Click the **Actions** drop-down menu and choose **Use a different scheme**.
4. There are three ways you can select your issue type scheme. Select the radio button that is most relevant:
  - a. **Choose an 'existing issue type scheme'** If you know the name of your scheme (e.g. 'Development Issue Type Scheme'), you can immediately choose it from the list. You will see a preview of issue types that would be available for your project as well as the description of the scheme.
  - b. **Choose a scheme that is the 'same as an existing project'** Select this option if you do not know the name of the scheme you would like to use, but you do know the name of the project whose set of issue types you wish to use for the project you are editing. You will be prompted to select a project and the scheme that is currently associated with the selected project will be used for your project as well.
  - c. **Create a new scheme and associate with current project** Select this option if you cannot find any existing scheme that fits your needs and would like to quickly create a new scheme. Simply select the relevant issue types for your project and a new scheme will be created with the default name and order. You can edit the name, default value and order of the newly created scheme [later](#).
5. If after you make your changes there are any issues in the selected project that will have obsolete issue types, they will have to be migrated with the [Issue Type Migration Wizard](#).

## Using the Issue Type Migration Wizard

The Issue Type Migration Wizard allows you to migrate issues from an obsolete issue type to a valid issue type. The wizard will be triggered whenever an action (e.g. editing a project's issue type scheme) results in an issue type becoming obsolete (not available in the scheme).

The wizard is similar to the bulk move function, except that you can't change the project of the issues. The major steps are:

1. Overview provides a summary of the issues that will require migration
2. Choose Issue Type
3. Set new status
4. Set field values
5. Confirmation

Steps 2 to 4 will be repeated for each issue type that requires migration. After you have migrated all the issues you'll see a summary of changes that will occur. If you click the '**Confirm**' button, the wizard will migrate your issues to the new issue types and then complete your action.

# Defining priority field values

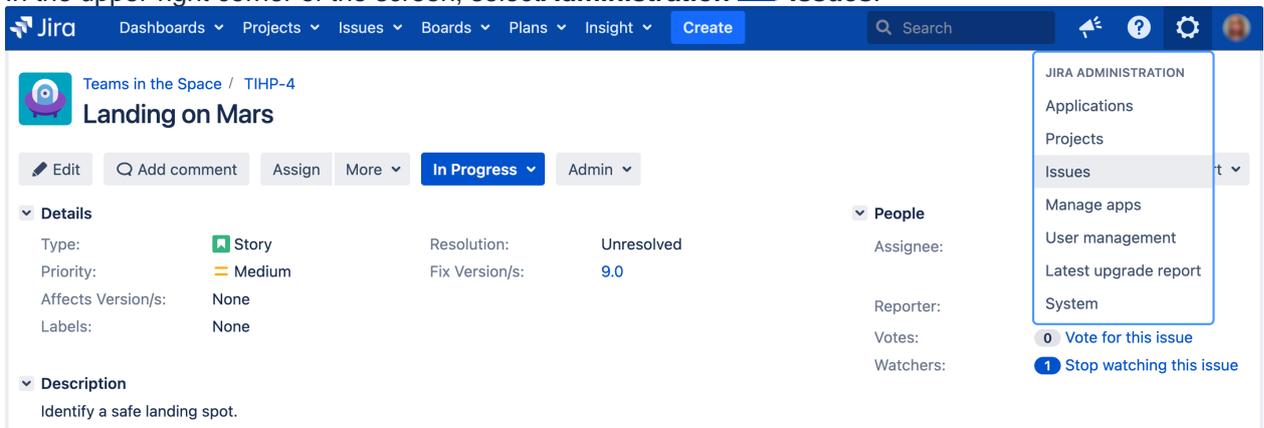
An issue's priority defines its importance in relation to other issues, so it helps your users determine which issues should be tackled first. Jira comes with a set of default priorities: Highest, High, Medium, Low, Lowest. You can modify these default priorities, create new ones, and add them to different projects by [associating these priorities with project priority schemes](#).

**i** For all of the following procedures, you must be logged in as a user with Jira administrator or Jira System Administrator [global permissions](#). For details on how permissions are set up in Jira, see [Permissions overview](#).

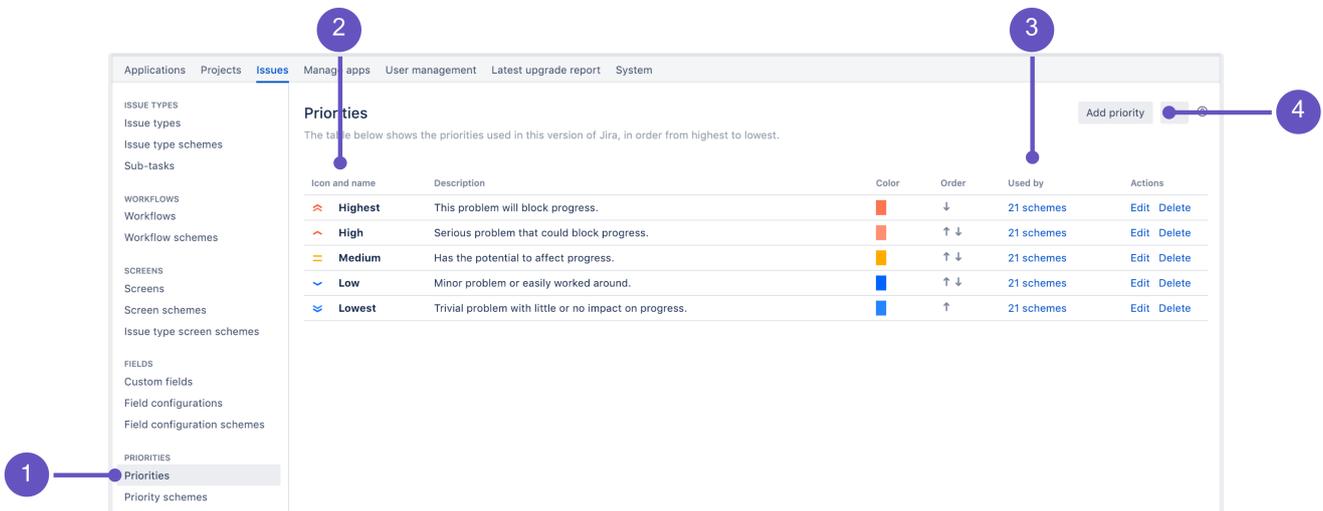
## Managing priorities

You can view and manage all issue priorities on the **Priorities** page:

1. In the upper-right corner of the screen, select **Administration**  **Issues**.



2. Select **Priorities** (the left-side panel):



1. **Priorities** tab that opens a list of all configured priorities.
2. **Icon and name** representing a priority.
3. Priority scheme that a priority is **used by**. You can click it to see the list of schemes.
4. Here you can create a new priority.

**✓** Use the JQL `priority` operator to quickly find and sort issues with a particular priority type by searching by priority name or priority ID. Check [Advanced searching - fields reference](#) for details.

## Creating priorities

To create a priority:

1. From the **Priorities** page, select **Add priority**.
2. Enter the name of your priority. The name will appear in the dropdown menu when a user creates or edits an issue.
3. Add a priority description (optional). The description won't appear in the issue view but can be helpful if you want to provide more context or share additional details with other users that have permission to manage priorities.
4. Choose an icon to represent this priority.
5. Specify a color to represent this priority. This color will be used by the Projects Dashboard Gadget to display the percentage of issues with different priority types within your project. You can either type the HTML color code or click the box at the right of the field to select from a color chart. See [Priority Color configuration will not change the color of a priority's text or icon](#) for details.
6. Select **Add** to create a priority. It will be added to the default priority scheme that contains all priorities. You can later add it to another scheme if you wish.

## Associating priorities with projects

To choose priorities for a project, you need to add them to a priority scheme, and then associate this scheme with a project. Until you do that, all projects use the default priority scheme. Treat priority schemes like mappings that allow you to choose a set of priorities and the projects that will use them. Read more about this here: [Associating priorities with projects](#).

## Translating priorities

You can translate priorities into different languages. See [Translating resolutions, priorities, statuses, and issue types](#).

## Editing and deleting priorities

To edit a priority, navigate to the **Priorities** page and then select **Edit** next to the priority you want to edit. To delete a priority, select **Delete** next to the priority you want to delete.

Priorities						
The table below shows the priorities used in this version of Jira, in order from highest to lowest.						
Icon and name	Description	Color	Order	Used by	Actions	
 <b>Highest</b>	This problem will block progress.		↓	1 scheme	<a href="#">Edit</a>	<a href="#">Delete</a>
 <b>High</b>	Serious problem that could block progress.		↑ ↓	1 scheme	<a href="#">Edit</a>	<a href="#">Delete</a>
 <b>Medium</b>	Has the potential to affect progress.		↑ ↓	1 scheme	<a href="#">Edit</a>	<a href="#">Delete</a>
 <b>Low</b>	Minor problem or easily worked around.		↑ ↓	1 scheme	<a href="#">Edit</a>	<a href="#">Delete</a>
 <b>Lowest</b>	Trivial problem with little or no impact on progress.		↑	1 scheme	<a href="#">Edit</a>	<a href="#">Delete</a>

 You can't delete priorities that are used by non-default priority schemes. You can see which schemes and how many of them are using a priority in the **Used by** column. To delete a priority, you first need to remove it from these schemes.

## Re-ordering priorities

Re-ordering priorities changes the order in which they appear in the dropdown menu when a user creates or edits an issue. The order on this page applies only to the default priority scheme.

To reorder priorities:

- Click the **up arrow** to move a priority higher up in the list.

- Click the **down arrow** to move a priority lower down in the list.

# Associating priorities with projects

Once you're happy with the priorities available in your Jira instance, it's time to associate them with some projects. You can choose a different set of priorities for each project by using priority schemes.

## What's a priority scheme?

A priority scheme works like a mapping that allows you to associate a subset of priorities with particular projects. You can use it to achieve the following goals:

- restrict the set of available priorities for a project,
- control the order in which priorities are displayed,
- select a default priority that is assigned to all newly created issues in a project.

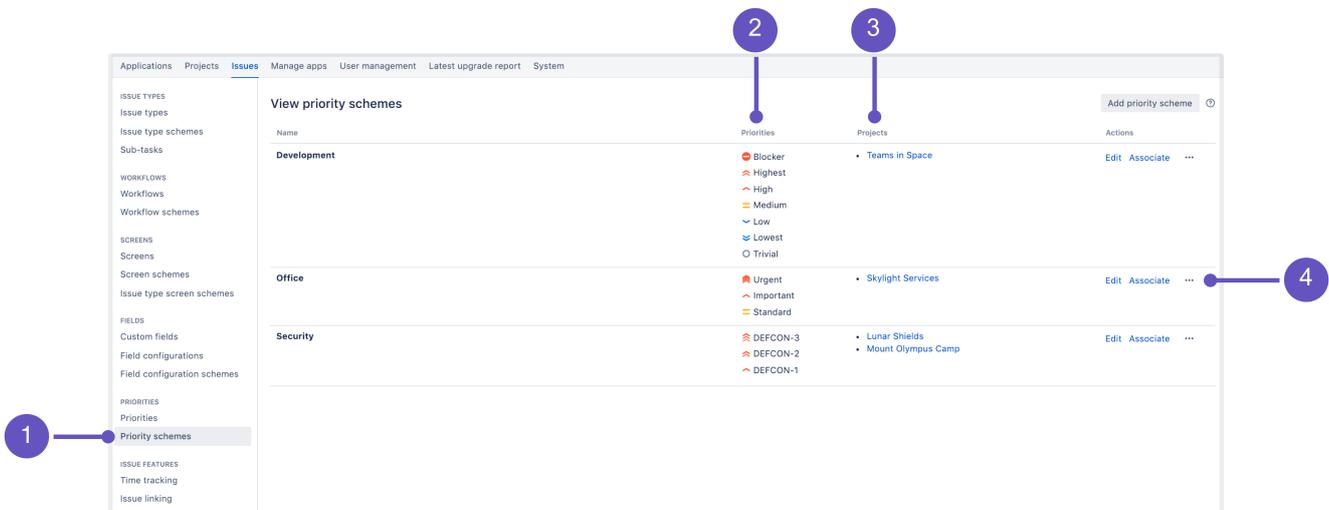
A single priority scheme can be reused across multiple projects so that a group of similar projects (i.e. projects which might be used for similar purposes) can share the same priorities. It's also easier to add or remove priorities for these projects because all you need to modify is a single priority scheme.

## Default priority scheme

Jira comes with a default priority scheme that contains all priorities, and is associated with all projects until you change it. You can't edit this scheme, but you can associate it as you wish.

## Managing priority schemes

To manage priority schemes and complete the actions listed below, go to **Administration** (⚙️) > **Issues**, and choose **Priority schemes**. You need to have the **Jira Administrators** global permission to get there.



1. This is where you can find all **priority schemes**.
2. **Priorities** used by a scheme.
3. A list of **projects** that use this set of priorities.
4. **Actions** that you can perform on a scheme: **Edit**, **Associate**, and **Delete**.

## Creating priority schemes

Managing priority schemes is difficult if you don't have any. Create a priority scheme, and decide which priorities apply to it.

To create a priority scheme:

1. From the Priority schemes page, select **Add priority scheme**.

2. Enter the name and description for your scheme. Make it meaningful, so that other admins know they can reuse this scheme instead of creating a new one.
3. Add priorities by dragging and dropping them from **Available priorities** to **Selected priorities**. The order on the list matters that's how your users will see priorities when assigning them to issues.
4. Select a default priority. It will be preselected whenever an issue is created in a project that uses this scheme.

## Associating priority schemes

Next, associate your priority scheme with a project.

1. From the Priority schemes page, select **Associate** next to the priority scheme you want to associate.
2. Select the projects that you want this scheme to apply to. If a project is already using a different scheme, it will switch to this one.
3. If some issues in the projects you've selected use priorities that are not available in this scheme, you'll be asked to choose priorities that will replace them. See [Replacing obsolete priorities](#).

## Other actions

Apart from creating and associating priority schemes, you can also edit and delete them.

Action	Description				
Editing	<p>To edit a priority scheme, select <b>Edit</b> next to the scheme you want to edit, and then change the name, default priority, or priorities selected for this scheme.</p> <p>If you removed priorities that are currently in use by some issues, you'll be asked to choose priorities that will replace them. See <a href="#">Replacing obsolete priorities</a>.</p>				
Deleting	<p>To delete a priority scheme, select <b>Delete</b> next to the scheme you want to delete. There are certain restrictions:</p> <ul style="list-style-type: none"> <li>• You can't edit, or delete the default priority scheme.</li> <li>• You can't delete priority schemes that are used by some projects. You can see how many projects are using a scheme on the Priorities scheme page (<b>Projects</b>). Associate these projects with a scheme (e.g. the default scheme), and proceed with deleting.</li> </ul>				
Ordering with JQL	<p>It's important to know how JQL works with priorities. If you use JQL to order issues by their priorities (e.g. descending), they will be ordered according to the importance of the priorities on the global list of priorities. The global list is what's displayed on the <a href="#">Priorities</a> page. The order from priority schemes is not relevant for JQL. To put it in an example, let's say the priorities in your Jira instance are arranged in the following way:</p> <table border="1" data-bbox="268 1473 1002 1720"> <thead> <tr> <th>Global list (Priorities page)</th> <th>Custom priority scheme</th> </tr> </thead> <tbody> <tr> <td> <ol style="list-style-type: none"> <li>1. Urgent</li> <li>2. Highest</li> <li>3. Medium</li> </ol> </td> <td> <ol style="list-style-type: none"> <li>1. Highest</li> <li>2. Urgent</li> <li>3. Medium</li> </ol> </td> </tr> </tbody> </table> <p>After appending a JQL query with e.g. <code>ORDER BY priority DESC</code>, the Urgent issues will be at the top of the list (followed by Highest, and then Medium), even if they're in a project that uses the custom scheme. You can change the order in the default scheme on the <a href="#">Priorities</a> page.</p>	Global list (Priorities page)	Custom priority scheme	<ol style="list-style-type: none"> <li>1. Urgent</li> <li>2. Highest</li> <li>3. Medium</li> </ol>	<ol style="list-style-type: none"> <li>1. Highest</li> <li>2. Urgent</li> <li>3. Medium</li> </ol>
Global list (Priorities page)	Custom priority scheme				
<ol style="list-style-type: none"> <li>1. Urgent</li> <li>2. Highest</li> <li>3. Medium</li> </ol>	<ol style="list-style-type: none"> <li>1. Highest</li> <li>2. Urgent</li> <li>3. Medium</li> </ol>				

## Replacing obsolete priorities

When you edit an existing scheme or associate a scheme with projects that were already using a different one, you might be asked to replace obsolete priorities. That's because some issues are using priorities that are not

available in the new or edited scheme. An issue can't live without a priority, so you need to replace the old with the new. It's simply about choosing e.g. **Highest** (new scheme) to replace **Top** (current scheme). Once you select priorities, we'll update all these issues for you.

The screenshot shows the 'Associate priority scheme' configuration page in Jira. The page has a navigation bar at the top with tabs for Applications, Projects, Add-ons, User Management, Issues (selected), System, and Audit Log. On the left, there is a sidebar menu with categories: ISSUE TYPES (Issue types, Issue type schemes, Sub-tasks), WORKFLOWS (Workflows, Workflow schemes), and SCREENS (Screens). The main content area is titled 'Associate priority scheme' and contains the following text: 'Some issues in the projects you've selected use priorities that are not available in this scheme. Select priorities that will replace them, and we'll update these issues for you.' Below this text is a table with three columns: 'Current scheme', 'New scheme', and 'Projects'. The 'Current scheme' column shows 'Default scheme' and 'Top'. The 'New scheme' column shows 'Space priorities' and a dropdown menu labeled 'Select new priority'. The 'Projects' column lists 'Teams in Space' and 'Teams on Mars'. At the bottom right of the page, there are 'Cancel' and 'Next' buttons.

Current scheme	→ New scheme	Projects
Default scheme	→ Space priorities	• Teams in Space • Teams on Mars
🚩 Top	→ <input type="text" value="Select new priority"/>	

# Defining resolution field values

Resolutions are the ways in which an issue can be closed. Jira applications ship with a set of default resolutions, but you can add your own as follows.

**i** For all of the following procedures, you must be logged in as a user with the **Jira Administrators** global permission.

## Defining a new resolution

### **!** Don't create a Resolution named "Unresolved"/"None"

Any issue that has the Resolution field set is treated by Jira applications as "resolved". The Issue Navigator displays Unresolved when no resolution is set for an issue. So adding a resolution named Unresolved/None and setting it in an issue will mean that the issue is seen as resolved. This will lead to confusion and is not recommended.

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Resolutions** to view the existing resolutions, along with a form for adding new resolutions:

Name	Description	Order	Operations
Done	Work has been completed on this issue.	↓	Edit - Delete - Default
Won't Do	This issue won't be actioned.	↑ ↓	Edit - Delete - Default
Duplicate	The problem is a duplicate of an existing issue.	↑	Edit - Delete - Default

Add New Resolution ⓘ

Name

Description

3. Complete the **Add New Resolution** form at the bottom of the page by entering the following details:
  - **Name** enter a short phrase that best describes your new resolution.
  - **Description** enter a sentence or two to describe when this resolution should be used.

**i** The **View Resolutions** page can be used to edit, delete, set as default, and re-order the resolutions as they are displayed to the user who is resolving an issue.

# Defining status field values

Statuses are used to represent the position of the issue in its [workflow](#). A workflow represents a business process, represented as a set of stages that an issue goes through to reach a final stage (or one of the final stages). Each stage in the workflow (called a *workflow step*) is linked to an *issue status*, and an issue status can be linked to only one workflow step in a given workflow.

Jira applications ships with a set of [default statuses](#) that are used by the [default workflow](#). You can add your own statuses and [customize](#) the workflow. You can also re-order existing statuses, as well as change their names, descriptions and lozenges.

## On this page:

- [Defining a new status](#)
- [Re-ordering statuses](#)
- [Deleting a status](#)

 For all of the following procedures, you must be logged in as a user with the [Jira Administrators global permission](#).

## Defining a new status

1. Choose **Administration**  > **Issues**.
2. Select **Statuses** to open the 'Statuses' page.
3. Click **Add Status** and complete the 'Add Status' form:
  - **Name** specify a short phrase that best describes your new status.
  - **Description** add a sentence or two to describe what workflow step this status represents.
  - **Category** choose a category that this status will be grouped into: 'To Do' (grey), 'In Progress' (blue) or 'Done' (green). Categories help you identify where issues are in their lifecycle, particularly in places where a large number of issues are rolled up, e.g. Version Details page, Sprint Health Gadget. The category is also used to map statuses to columns in Jira Software, when creating a new board for an existing project.

### Next steps:

Now you will need to associate your new status with a workflow 'step'. See [Working with workflows](#).

## Re-ordering statuses

You may want to change the order of statuses in Jira in line with a particular workflow or to highlight key statuses. The order of statuses is reflected on screens (or parts of the screen) in Jira, where issues are listed or grouped by status. These include the issues summary for a project, search results (when status is one of the columns), and a number of gadgets, like the Issue Statistics gadget (where the Statistic Type is 'Status').

1. Navigate to the 'Statuses' page (described in the 'Defining a new status' section above).
2. Use the up and down arrows in the **Order** column to re-order individual statuses.

## Deleting a status

You can only delete statuses that are not being used in workflows, i.e. inactive statuses.

1. Navigate to the 'Statuses' page (described in the 'Defining a new status' section above).
2. Click **Delete** for the status that you want to delete.

# Translating resolutions, priorities, statuses, and issue types

Further extending Jira as an international issue manager, it is possible to easily specify a translated name and description for all values of the following 'issue constants':

- the [issue type](#) field (for either standard and sub-task issue types)
- the [status](#) field
- the [resolution](#) field
- the [priority](#) field

This allows you to specify a translation set for each available language providing each user with a more complete translation in their own chosen language. The translated field names and descriptions appear throughout Jira, e.g. in reports, gadgets and all issue views.

## Translating an issue constant

Each issue constant can be configured to have a translation set for each available language in your Jira system. If no translation has been configured for a particular language, the default issue constant name and description are displayed.

1. Log in as a user with the [Jira Administrators global permission](#).
2. Click the **Translate** link located on any of the following pages:
  - [Defining issue type field values](#) (for standard issue types - click any of the **Translate** links),
  - [Configuring sub-tasks](#) (for sub-task issue types),
  - [View statuses](#),
  - [View resolutions](#), or
  - [View priorities](#).

The relevant issue constant **Translation** page displays the translation set for the currently selected language.

3. To view/update a translation set for a specific language, select the required language from the **View Language Translations** list at the top of the page and click the **View** button to preview the translation:

Sub-Task Translations

• View sub-tasks

On this page, you can add 'Sub-Task' translations for the installed languages. Any translations you define here will override any translations that may exist for the issue constants in your languages resource bundle. To revert to the resource bundle values just set the name/description pair to blank.

View Language Translations: French (France) View

Sub-Task Translation Language: English (United States)

Sub-Task	Name	Description
Concepts Collection of ideas in wireframe format	Concepts	Collection d'idées en wirefram
Designs Collection of polished designs in pixel perfect format	Designs	Collection de designs peaufinés
Documentation SubTask		

Note that a translated name and description set can be specified for each type of issue constant.

4. Once all translations have been entered, select **Update**. Note that:
  - The process can be repeated for all of the issue constants i.e. Issue Type, Status, Resolution and Priority fields.
  - The translated issue constant name and description will be displayed throughout Jira, e.g. in reports, gadgets and all issue views.

**i** The default issue constant name and description are displayed if a translation has not been specified.

# Issue fields and statuses

These are the pieces that make up the issues you work on:

- [Issue fields](#)
- [Issue types](#)
- [Issue priorities](#)
- [Issue resolutions](#)

## Issue fields

Field	Description
<b>Project</b>	The parent project to which the issue belongs.
<b>Key</b>	A unique identifier for this issue, in the example above: ANGRY-304. (The characters to the left of the hyphen represent the project to which this issue belongs.)
<b>Summary</b>	A brief one-line summary of the issue. For example, "Red Angry Nerd is scary."
<b>Type</b>	See below for a list of types.
<b>Status</b>	The stage the issue is currently at in its lifecycle ( <a href="#">workflow</a> ). See below for a list of statuses.
<b>Priority</b>	The importance of the issue in relation to other issues. (See below for a list of priorities).
<b>Resolution</b>	A record of the issue's <a href="#">resolution</a> , if the issue has been resolved or closed. (See below for a <a href="#">list of resolutions</a> ).
<b>Affects Version(s)</b> <i>(if applicable)</i>	Project <a href="#">version(s)</a> for which the issue is (or was) manifesting.
<b>Fix Version(s)</b> <i>(if applicable)</i>	Project <a href="#">version(s)</a> in which the issue was (or will be) fixed.
<b>Component(s)</b> <i>(if applicable)</i>	Project <a href="#">component(s)</a> to which this issue relates.
<b>Labels</b> <i>(if applicable)</i>	Labels to which this issue relates.
<b>Environment</b> <i>(if applicable)</i>	The hardware or software environment to which the issue relates.

<b>Description</b>	A detailed description of the issue.
<b>Links</b>	A list of links to related issues. (Strikethrough text, <del>like this</del> , indicates that an issue has been resolved.)
<b>Assignee</b>	The person to whom the issue is currently assigned. Note that you cannot assign issues to a user group.
<b>Reporter</b>	The person who entered the issue into the system.
<b>Votes</b>	The number shown indicates how many votes this issue has.
<b>Watchers</b>	number shown indicates how many people are watching this issue.
<b>Due</b> (if applicable)	The date by which this issue is scheduled to be completed.
<b>Created</b>	The time and date on which this issue was entered into Jira.
<b>Updated</b>	The time and date on which this issue was last edited.
<b>Resolved</b>	The time and date on which this issue was resolved.
<b>Estimate</b>	The <b>Original Estimate</b> of the total amount of time required to resolve the issue, as estimated when the issue was created.
<b>Remaining</b>	The <b>Remaining Estimate</b> , i.e. the current estimate of the remaining amount of time required to resolve the issue.
<b>Logged</b>	The sum of the <b>Time Spent</b> from each of the individual work logs for this issue.
<b>Development *</b>	If you use Bitbucket to manage your code repositories, you can create code branches in your code development tools directly from Jira issues. See <a href="#">Integrating with development tools</a> for details.
<b>Agile *</b>	Lets you view your issue on your Scrum or Kanban board.
<b>Service Desk **</b>	Lets you view request participants and view the equivalent request in the customer portal

\*Only available in Jira Software

projects, and only available to Jira Software users

\*\* Only available in Jira Service Management projects, and only available to Jira Service Management users

## Issue types

Your default issue types depend on what Jira application you have installed. We've listed all the default issue types for each application:

Type	Description
Task	A task represents work that needs to be done.
Sub-task	A sub-task is a piece of work that is required for a task.

Type	Description
Task	A task represents work that needs to be done.
Sub-task	A sub-task is a piece of work that is required for a task.
Story	A user story is the smallest unit of work that needs to be done.
Bug	A bug is a problem which impairs or prevents the functions of a product.
Epic	A big user story that needs to be broken down.

Type	Description
IT Help	Requesting help for IT related problems.
Purchase	Requesting hardware or software.
Change	Requesting a change in current IT profile.
Fault	Reporting a fault.
Access	Requesting additional access.

### Issue priorities

An issue's priority indicates its relative importance. The default priorities are listed below; note that both the priorities and their meanings can be customized by your administrator to suit your organization.

Priority	Description
Highest	Highest priority. Indicates that this issue takes precedence over all others.
High	Indicates that this issue is causing a problem and requires urgent attention.
Medium	Indicates that this issue has a significant impact.
Low	Indicates that this issue has a relatively minor impact.
Lowest	Lowest priority.

### Issue resolutions

An issue can be completed, or resolved, in many ways. An issue resolution is usually set when the status is changed. The default resolutions are listed below; note that your administrator may have customized these to suit your organization.

Resolution	Description
Done	The work is completed.
Won't do	The work will not be done.
Duplicate	This work is being tracked elsewhere.

Resolution	Description
Done	The work is completed.
Won't do	The work will not be done.

Duplicate	This work is being tracked elsewhere.
Cannot reproduce	The issue cannot be reproduced.

Resolution	Description
Done	The work is completed.
Won't do	The work will not be done.
Duplicate	This work is being tracked elsewhere.

Note that once an issue has been resolved (that is, the issue's Resolution field is filled in), textual references to that issue will show the key in strikethrough text.

# Configuring issue-level security

Issue security levels are created within issue security schemes and let you control which user or group of users can view an issue. When an issue security scheme is associated with a project, its security levels can be applied to issues in that project. Sub-tasks will also inherit the security level of their parent issue.

Note, if issue security levels are available but aren't set, the project permissions will then be applied.

## On this page:

- [Before you begin](#)
- [Creating an issue security scheme](#)
- [Assigning an issue security scheme to a project](#)
- [Deleting an issue security scheme](#)
- [Editing an issue security scheme](#)
- [Copying an issue security scheme](#)

## Before you begin

- Log in as a user with the [Jira Administrators global permission](#) to configure issue-level security.
- Make sure all users who want to use issue-level security have the project-specific ['Set Issue Security' permission](#).

## Creating an issue security scheme

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Issue Security Schemes** to open the Issue Security Schemes page.
3. Click **Add Issue Security Scheme**.
4. Fill in the requested details and click **Add**.

## Adding a security level to an issue security scheme

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Issue Security Schemes** to open the Issue Security Schemes page.
3. Click the scheme name, or the **Security Levels** link in the Actions column, to open the Edit Issue Security Levels page.
4. Fill in the requested details and then click **Add Security Level**.

## Setting the default security level for an issue security scheme

You now have the power to select the default security level that will be applied to issues assigned to each security scheme. Some things to keep in mind when setting a default security level:

- If the reporter of an issue does not have the ['Set Issue Security' permission](#), the issue will be set to the default security level.
- If an issue security scheme doesn't have a default security level, issue security levels will be set to 'None' (anyone can see the issues).

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Issue Security Schemes** to open the Issue Security Schemes page.
3. Click the scheme name, or the **Security Levels** link in the Actions column, to open the Edit Issue Security Levels page.
  - a. To set the default security level, locate the appropriate **Security Level** and click **Default** in the Actions column.
  - b. To remove the default security level, click **Change default security level to "None"** link (near the top of the page).

## Adding members to a security level

1. Choose **Administration**  > **Issues**.
2. Select **Issue Security Schemes** to open the Issue Security Schemes page.
3. Click the name of any scheme or the link **Security Levels** to open the **Edit Issue Security Levels** page.
4. Locate the appropriate security level and click its **Add** link (in the **Actions** column), which opens the **Add User/Group/Project Role to Issue Security Level** page.
5. Select the appropriate user, group or project role, then click the **Add** button.

 A security level's members may consist of:

- Individual users
- Groups
- [Project roles](#)
- Issue roles such as 'Reporter', 'Project Lead', and 'Current Assignee'
- 'Anyone' (eg. to allow anonymous access)
- A (multi-)user or (multi-)group picker [custom field](#).

6. Repeat steps 4 and 5 until all appropriate users, groups, or project roles have been added to the security level.

## Assigning an issue security scheme to a project

1. Choose **Administration**  > **Projects**, and select the relevant project.
2. Select the name of the project of interest. The **Project Summary** page is displayed.
3. In the **Permissions** section of the **Project Summary** page, click the link corresponding to the **Issues** label to open the **Associate Issue Security Scheme to Project** page. This will either be the name of the project's current issue security scheme, or the word **None**.
4. Select the issue security scheme that you want to associate with this project.
5. If there are no previously secured issues (or if the project did not previously have an issue security scheme), skip the next step.
6. If there are any previously secured issues, select a new security level to replace each old level. All issues with the security level from the old scheme will now have the security level from the new scheme. You can choose 'None' if you want the security to be removed from all previously secured issues.
7. Click the **Associate** button to associate the project with the issue security scheme.

 If the **Security Level** field is not displayed on the issue's screen after configuring the Issue-Level Security, use the Where is My Field? tool to see why it is not being displayed.

If the **Security Level** field has been hidden on purpose, please see the limitations of doing so in [Hiding or showing a field](#).

## Deleting an issue security scheme

It's important to understand that you can't delete a issue security scheme if it is associated with a project. You must first remove any associations between the issue security scheme and projects in your Jira installation please refer to [Assigning an Issue Security Scheme](#).

1. Choose **Administration**  > **Issues**.
2. Select **Issue Security Schemes** to open the Issue Security Schemes page, which lists all the issue security schemes currently available in your Jira installation.
3. Click the **Delete** link (in the **Actions** column) for the scheme that you want to delete.
4. On the confirmation page, click **Delete** to confirm the deletion. Otherwise, click **Cancel**.

## Editing an issue security scheme

You can edit the name and description of an issue security scheme. You can also edit the Default Security Level when editing an issue, and the security level will be applied in the same manner as described in [Setting the default security level for an issue security scheme](#).

1. Choose **Administration**  > **Issues**.
2. Select **Issue Security Schemes** to open the Issue Security Schemes page, which lists all the issue security schemes currently available in your Jira installation.
3. Click the **Edit** link (in the **Actions** column) for the scheme that you want to edit.
4. Make your edits, and then click **Update** to confirm the edits. Otherwise, click **Cancel**.

### Copying an issue security scheme

1. Choose **Administration**  > **Issues**.
2. Select **Issue Security Schemes** to open the Issue Security Schemes page, which lists all the issue security schemes currently available in your Jira installation.
3. Click the **Copy** link (in the **Actions** column) for the scheme that you want to copy. A new scheme will be created with the same security levels and the same users/groups/project roles assigned to them. Your new scheme will be called '**Copy of ...**'. You can edit your new scheme to give it a different name if you wish.

# Configuring permissions

When configuring security for your Jira application instance, there are two areas to address:

- permissions within Jira applications themselves
- security in the external environment

## Configuring permissions within Jira applications

Jira applications have a flexible security system which allows you to configure who can access Jira applications, and what they can do/see within them.

There are five types of security levels within Jira applications:

1. [Global permissions](#) these apply to Jira applications as a whole.
2. [Project permissions](#) organized into permission schemes, these apply to projects as a whole (e.g. who can see the project's issues ('Browse' permission), create, edit and assign them).
3. [Issue security levels](#) organized into security schemes, these allow the visibility of individual issues to be adjusted, within the bounds of the project's permissions.
4. [Comment visibility](#) allows the visibility of individual comments (within an issue) to be restricted.
5. [Work-log visibility](#) allows the visibility of individual work-log entries (within an issue) to be restricted. Does not restrict visibility of progress bar on issue time tracking.

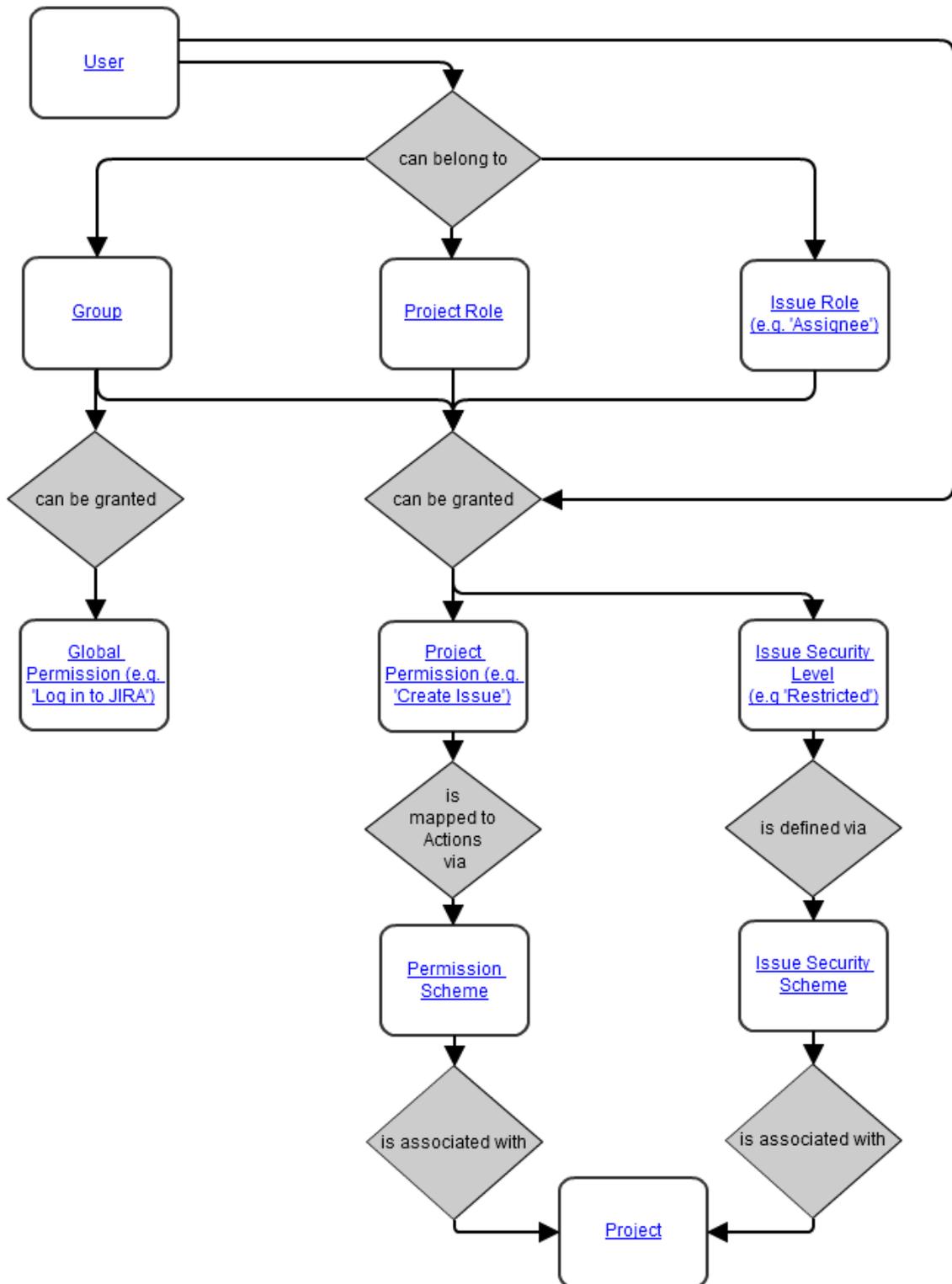
### On this page:

- [Configuring permissions within Jira applications](#)
- [Configuring security in the external environment](#)

### In this section:

- [Managing global permissions](#)
- [Managing project permissions](#)
  - [Customizing Jira Service Management permissions](#)
  - [Resolving Jira Service Management permission errors](#)
  - [Using Manage Sprints permission for advanced cases](#)
  - [Sprint permissions and defined processes](#)
- [Managing project roles](#)
  - [Managing project role membership](#)
- [Allowing anonymous access to your project](#)

## Diagram: People and permissions



## Configuring security in the external environment

If your Jira application instance contains sensitive information, you may want to configure security in the environment in which your instance is running. Some of the main areas to consider are:

- File system you should restrict access to the following directories (but note that the user which your instance is running as will require full access to these directories):
  - [Index directory](#)
  - [Attachments directory](#)

- Database:
  - If you are using an [external database](#) as recommended for production systems (i.e. you are not using Jira's internal/bundled H2 database), you should restrict access to the database that your Jira instance uses.
  - If you are using Jira's internal/bundled H2 database, you should restrict access to the directory in which you [installed](#) Jira. (Note that the user which your Jira instance is running as will require full access to this directory.)
- SSL if you are running your Jira instance over the Internet, you may want to consider using [SSL](#).

# Managing global permissions

Global permissions are system-wide and are granted to groups of users. You can refer to [project permissions](#) to manage permissions that apply to individual projects.

 For all of the following procedures, you must be logged in as a user with the **Jira Administrators** or **Jira System Administrators** global permission.

## On this page:

- [Granting global permissions](#)
- [Removing global permissions](#)
- [About Jira System Administrators and Jira Administrators](#)
- [Separating Jira System Administrators from Jira Administrators in default Jira installations](#)
- [Troubleshooting permissions with the Jira admin helper](#)

This table lists the different global permissions and the functions they secure:

Global Permission	Explanation
<b>Jira System Administrators</b>	Permission to perform all Jira administration functions.
<b>Jira Administrators</b>	Permission to perform most Jira administration functions. Note that a user with the <b>Jira Administrators</b> permission will be able to log in at any time, but may have restricted functions depending on their application access.
<b>Browse Users</b>	Permission to view a list of all Jira user names and group names, share issues, and @mention people on issues. Used for selecting users/groups in popup screens. Enables auto-completion of user names in most 'User Picker' menus and popups. Note that the <a href="#">Assign User</a> permissions also allows a limited version of this on a per-project basis.
<b>Create Shared Objects</b>	Permission to share a filter or dashboard globally or with groups of users. Also used to control who can create an agile board.
<b>Manage Group Filter Subscriptions</b>	Permission to manage (create and delete) group filter subscriptions.

<b>Bulk Change</b>	<p>Permission to execute the bulk operations within Jira:</p> <ul style="list-style-type: none"> <li>- Bulk Edit *</li> <li>- Bulk Move *</li> <li>- Bulk Workflow Transition</li> <li>- Bulk Delete *</li> </ul> <p>(* subject to <a href="#">project-specific permissions</a>.)</p> <p>The decision to grant the <b>Bulk Change</b> permission should be considered carefully. This permission grants users the ability to modify a collection of issues at once. For example, in Jira installations configured to run in <b>Public</b> mode (i.e. anybody, even people from outside of your organization) can sign up and create issues), a user with the <b>Bulk Change</b> global permission and the <b>Add Comments</b> project permission could comment on <i>all</i> accessible issues. Undoing such modifications may not be possible through the Jira application interface and may require changes made directly against the database (which is not recommended).</p>
--------------------	--

## Granting global permissions

1. In the upper-right corner of the screen, select **Administration**  > **System**.
2. Select **Global Permissions** to open the Global Permissions page, which lists Jira's global permissions.

The **Add Permission** box is shown at the bottom of the list (not displayed in the screen capture above).

3. In the **Permission** drop-down list, select the global permission you wish to grant.
  4. In the **Group** drop-down list, either:
    - select the group to which you wish to grant the permission; or
    - if you wish to grant the permission to non logged-in users, select **Anyone on the web**. This is **not** recommended for production systems, or systems that can be accessed from the public Internet such as Cloud.

**Please Note:**

    - If you have reached your user limit, you will be able to create new users but it won't have login permission.
- Jira admin doesn't consume a license unless they've been granted specific Jira application access. See [Licensing and application access](#).

## Removing global permissions

1. In the upper-right corner of the screen, select **Administration**  > **System**.
2. Select **Global Permissions** to open the Global Permissions page, which lists Jira's global permissions.
3. For each global permission in Jira (indicated on the left of this page), groups which currently have that permission are shown on the right (under the **Users / Groups** column).
4. Locate the global permission you want to remove from a group as well as the group you want to remove that permission from (under **Users / Groups**) and click the **Delete** link next to that group.

## About Jira System Administrators and Jira Administrators

People who have the **Jira System Administrators** permission can perform all of the administration functions in Jira, while people who have only the **Jira Administrators** permission cannot perform functions which could affect the application environment or network. This [separation](#) is useful for organizations which need to delegate some administrative privileges (e.g. creating users, creating projects) to particular people, without granting them complete rights to administer the Jira system.

Here is a list of administration tasks that only **Jira System Administrators** (not **Jira Administrators**) can perform:

- View or manage tasks from the [Systems menu](#).
- Configure Jira's [SMTP mail server](#) for notifications (but *they can* configure [POP/IMAP mail servers](#) for the receipt of email messages that create issue comments and new issues, and fully administer [email notification schemes](#)).
- Configure a CVS source code repository (but *they can* associate a [project](#) with a configured repository).
- Configure [listeners](#).
- Configure [services](#) (except for [POP/IMAP](#) services).
- Configure [issue cloning](#).
- Change the [index path](#) (but *they can* reindex and optimize the index).
- Run the [integrity checker](#).
- Access [logging and profiling](#) information.
- Access the scheduler.
- [Export/backup](#) Jira data to XML.
- [Import/restore](#) Jira data from XML.
- Import [XML workflows](#) into Jira.
- Configure [attachments](#) (note that **Jira Administrators** can set the size limits of attachments, enable thumbnails, and enable ZIP support).
- Add gadgets to the [gadget directory](#).
- Configure [user directories](#) (e.g. LDAP).
- Configure [Application Links](#) that use an authentication type other than OAuth.
- View [user sessions](#).
- Access [license details](#).
- Grant/revoke the **Jira System Administrators** global permission.
- Edit (or Bulk Edit) [groups](#) that have the **Jira System Administrators** global permission.
- Edit, change the password of or delete a user who has the **Jira System Administrators** global permission.
- Upload and/or install an [app](#).
- Configure [an announcement banner](#).

It is recommended that people who have the **Jira Administrators** permission (and not the **Jira System Administrators** permission) are not given direct access to the Jira filesystem or database.

## Separating Jira System Administrators from Jira Administrators in default Jira installations

By default, the `jira-administrators` group has both the **Jira Administrators** permission and the **Jira System Administrators** permission. Also by default, the user account created during the [Jira setup wizard](#) is a member of this `jira-administrators` group.

If you need some people to have only the **Jira Administrators** permission (and not the **Jira System Administrators** permission), you will need to use two separate groups, e.g.:

1. [Create a new group](#) (e.g. called `jira-system-administrators`).
2. Add to the `jira-system-administrators` group everyone who needs to have the **Jira System Administrators** permission.
3. Grant the **Jira System Administrators** permission to the `jira-system-administrators` group.
4. Remove the **Jira System Administrators** permission from the `jira-administrators` group.
5. (Optional, but recommended for ease of maintenance) Remove from the `jira-administrators` group everyone who is a member of the `jira-system-administrators` group.

## Troubleshooting permissions with the Jira admin helper

The Jira admin helper can help you diagnose why a user can or cannot see a certain issue.

**Note:** For all of the following procedures, you must be logged in as a user with the **Jira Administrators** global permission.

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. Select **Permission helper**.
3. Enter the username of the user (leave blank for anonymous users), an issue key (for example, an issue that the user can/cannot see) and the permission to check.
4. Click **Submit**.

# Managing project permissions

Project permissions are created within [permission schemes](#), which are then assigned to specific projects by Jira Administrators. Project permissions are able to be granted based on:

- Individual users
- Groups
- [Project roles](#)
- Issue roles such as 'Reporter', 'Project Lead', and 'Current Assignee'
- 'Anyone' (e.g. to allow anonymous access)
- A (multi-)user picker [custom field](#).
- A (multi-)group picker [custom field](#). This can either be an actual group picker custom field, or a (multi-)select-list whose values are group names.

Note that some permissions are dependent upon others to ensure that users can perform the actions needed. For example, in order for a user to be able to resolve an issue, that user must be granted both the Transition Issue permission and the Resolve Issue permission.

## On this page:

- [Permission schemes](#)
- [Creating a permission scheme](#)
- [Adding users, groups, or roles to a permission scheme](#)
- [Deleting users, groups, or roles from a permission scheme](#)
- [Associating a permission scheme with a project](#)
- [Deleting a permission scheme](#)
- [Copying a permission scheme](#)

The following table lists the different types of project permissions and the functions they secure. Note that project permissions can also be used in [workflow conditions](#).

## Project permissions overview

Project permissions	Explanation
Administer projects	<p>Permission to administer a project in Jira. This includes the ability to edit <a href="#">project role membership</a>, <a href="#">project components</a>, <a href="#">project versions</a>, and some <a href="#">project details</a> ('Project Name', 'URL', 'Project Lead', 'Project Description').</p> <p>This permission granted together with the 'Browse projects' allows you to see the audit log for a specific project.</p>
Extended project administration	<p>Gives the project administrator the ability to edit workflows and screens under certain conditions.</p> <ul style="list-style-type: none"> <li>• The workflow must not be shared with any other projects, or be a system workflow.</li> <li>• To add a status, the status must already exist in the Jira instance i.e. the project admin can't create new statuses or edit existing statuses.</li> <li>• To delete a status, the status must not be used by any of the project's issues.</li> <li>• The project admin can create, update or delete transitions, but they can't select or update a screen used by the transition, or edit or view a transition's properties, conditions, validators or post-functions.</li> <li>• The screen must not be a default system screen.</li> <li>• The screen must not be shared with any other projects, or used as a transition screen in workflows.</li> <li>• The project admin can add, remove and rearrange system fields.</li> <li>• The project admin can add, remove and rearrange existing custom fields, but they cannot create custom fields.</li> </ul>

Browse projects	<p>Permission to browse projects, use the Issue Navigator and view individual issues (except issues that have been restricted via <a href="#">issue-level security</a>). <b>Many other permissions are dependent on this permission</b>, e.g. the 'Work On Issues' permission is only effective for users who also have the 'Browse Projects' permission.</p> <p>This permission granted together with the 'Administer projects' allows you to see the audit log for a specific project.</p>
Manage sprints (only available to Jira Software users)	<p>Permission to perform the following sprint-related actions for all projects in a board:</p> <ul style="list-style-type: none"> <li>• Creating sprints</li> <li>• Starting sprints</li> <li>• Completing sprints</li> <li>• Reopening sprints</li> <li>• Reordering future sprints</li> <li>• Adding sprint goals</li> <li>• Deleting sprints</li> <li>• Editing sprint information (sprint name, goal, dates)</li> <li>• Moving the sprint footer</li> </ul> <p>Depending on the complexity of your board's filter query, you may need further consideration when configuring the 'Manage Sprints' permission for users. For more information on the impact of complex filters, and ways to simplify your filter query, see <a href="#">Using Manage Sprints permission for advanced cases</a>.</p> <p>In general, sprint actions require the Manage Sprints permission. However, there are some sprint actions (e.g. adding issues to sprints, removing issues from sprints) that require the Schedule Issues and Edit Issues permissions.</p> <p>When adding an issue to a sprint:</p> <ul style="list-style-type: none"> <li>• Sub-tasks cannot be moved independently of their parents.</li> <li>• An issue can only be assigned to one active sprint or future sprint. This means you can't add an issue to both an active sprint and a future sprint at the same time.</li> <li>• You can add any issue to any active or future sprint, even if the issue doesn't match the filter query of the board where the sprint was created. When you do this: <ul style="list-style-type: none"> <li>• the issue is assigned to the sprint, but will not be visible on boards where the filter query excludes it.</li> <li>• any sprint actions (e.g. start sprint, close sprint) that span multiple boards will also affect the sprint in all boards where the sprint is visible.</li> <li>• if the issue doesn't match the filter query of any agile board, the issue will be linked to the sprint but not appear in any board.</li> </ul> </li> <li>• A sprint appears in any board a single board or multiple boards as long as the issues assigned to the sprint match the filter query of the board or boards. This also applies to completed sprints.</li> </ul> <p>See <a href="#">Planning sprints</a> for more information.</p>
Start/Complete sprints (only available to Jira Software users)	<p>Permission to start sprints and end them when the sprint dates are set. This permission doesn't allow for changing any sprint properties, such as the name, goal, and dates. You can only change sprint status to Active or Completed.</p>
Edit sprints (only available to Jira Software users)	<p>Permission to change the sprint <i>name and goal</i>. With this permission you can't change sprint dates or start or end a sprint.</p>

View development tools (only available to Jira Software users)	Permission to <a href="#">view the Development panel</a> , which provides you with just enough information to evaluate the status of an issue's development, at a glance.
View (read-only) workflow	Permission to view the project's 'read-only' <a href="#">workflow</a> when viewing an issue. This permission provides the 'View Workflow' link against the 'Status' field of the 'View Issue' page.
<b>Issue permissions</b>	<b>Explanation</b>
Assign issues	Permission to assign issues to users. Also allows autocompletion of users in the Assign Issue dropdown. (See also Assignable User permission below)
Assignable user	Permission to be assigned issues. (Note that this does not include the ability to assign issues; see Assign Issue permission above).
Close issues	Permission to <a href="#">close</a> issues based on the workflow conditions. (This permission is useful where, for example, developers resolve issues and testers close them). Requires the Transition issue and Resolve issue transitions. Also see the Resolve Issues permission.
Create issues	Permission to <a href="#">create</a> issues in the project. (Note that the Create Attachments permission is required in order to create attachments.) Includes the ability to <a href="#">create sub-tasks</a> (if sub-tasks are <a href="#">enabled</a> ).
Delete issues	Permission to delete issues. Think carefully about which groups or project roles you assign this permission to; usually it will only be given to administrators. Note that deleting an issue will delete all of its comments and attachments, even if the user does not have the Delete Comments or Delete Attachments permissions. However, the Delete Issues permission does not include the ability to delete individual comments or attachments.
Edit issues	Permission to edit issues (excluding the 'Due Date' field see the Schedule Issues permission). Includes the ability to convert issues to <a href="#">sub-tasks</a> and vice versa (if sub-tasks are <a href="#">enabled</a> ). Note that the Delete Issue permission is required in order to delete issues. The Edit Issue permission is usually given to any groups or project roles who have the Create Issue permission (perhaps the only exception to this is if you give <b>every one</b> the ability to create issues it may not be appropriate to give everyone the ability to edit too).
Link issues	Permission to <a href="#">link</a> issues together. (Only relevant if Issue Linking is <a href="#">enabled</a> ).
Modify reporter	Permission to modify the 'Reporter' of an issue. This allows a user to create issues 'on behalf of' someone else. This permission should generally only be granted to administrators.
Move issues	Permission to <a href="#">move</a> issues from one project to another, or from one workflow to another workflow within the same project. Note that a user can only move issues to a project for which they have Create Issue permission.
Resolve issues	Permission to <a href="#">resolve</a> and reopen issues based on the workflow condition. This also includes the ability to set the 'Fix For version' field for issues. Requires the Transition issues permission. Also see the Close Issues permission.
Schedule issues	Permission to <a href="#">schedule</a> an issue that is, to edit the 'Due Date' of an issue. In older versions of Jira this also controlled the permission to view the 'Due Date' of an issue.
Set issues security	Permission to set the <a href="#">security level</a> on an issue to control who can access the issue. Only relevant if issue security has been <a href="#">enabled</a> .

Transition issues	Permission to <a href="#">transition</a> (change) the status of an issue.
<b>Voters &amp; watchers permissions</b>	<b>Explanation</b>
Manage watcher list	Permission to manage (i.e. view/add/remove users to/from) the watcher list of an issue.
View voters and watchers	Permission to view the voter list and watcher list of an issue. Also see the Manage Watcher List permission.
<b>Comments permissions</b>	<b>Explanation</b>
Add comments	Permission to add comments to issues. Note that this does not include the ability to edit or delete comments.
Delete all comments	Permission to delete any comments, regardless of who added them.
Delete own comments	Permission to delete comments that were added by the user.
Edit all comments	Permission to edit any comments, regardless of who added them.
Edit own comments	Permission to edit comments that were added by the user.
<b>Attachments permissions</b>	<b>Explanation</b>
Create attachments	Permission to attach files to an issue. (Only relevant if attachments are <a href="#">enabled</a> ). Note that this does not include the ability to delete attachments.
Delete all attachments	Permission to delete any attachments, regardless of who added them.
Delete own attachments	Permission to delete attachments that were added by the user.
<b>Time-tracking permissions</b>	<b>Explanation</b>
Work on issues	Permission to log work against an issue, i.e. create a worklog entry. (Only relevant if <a href="#">time tracking</a> is enabled).
Delete all worklogs	Permission to delete any worklog entries, regardless of who added them. (Only relevant if <a href="#">time tracking</a> is enabled). Also see the Work On Issues permission.
Delete own worklogs	Permission to delete worklog entries that were added by the user. (Only relevant if <a href="#">time tracking</a> is enabled). Also see the Work On Issues permission.
Edit all worklogs	Permission to edit any worklog entries, regardless of who added them. (Only relevant if <a href="#">time tracking</a> is enabled). Also see the Work On Issues permission.
Edit own worklogs	Permission to edit worklog entries that were added by the user. (Only relevant if <a href="#">time tracking</a> is enabled). Also see the Work On Issues permission.
<b>Archiving permissions</b>	<b>Explanation</b>

Archive issues for a project	Permission to archive issues in a specific project. This permission does not allow for archiving issues in bulk.
Restore issues for a project	Permission to restore issues to a specific project.
Browse archive	Permission to view all archived issues on the <b>Issues &gt; Archived issues page</b> .
Browse project archive	Permission to view archived issues that belong to a specific project. These issues are displayed on the <b>Issues &gt; Archived issues page</b> .

## Permission schemes

### What is a permission scheme?

A permission scheme is a set of user/group/role assignments for the project permissions listed above. Every project has a permission scheme. One permission scheme can be associated with multiple projects.

### Why permission schemes?

In many organizations, multiple projects have the same needs regarding access rights. (For example, only the specified project team may be authorized to assign and work on issues).

Permission schemes prevent having to set up permissions individually for every project. Once a permission scheme is set up it can be applied to all projects that have the same type of access requirements.

## Creating a permission scheme

1. Choose **Administration**  > **Issues**.
2. Select **Permission Schemes** to open the Permission Schemes page, which displays a list of all permission schemes in your Jira system and the projects that use each scheme.
3. Click the '**Add Permission Scheme**' link.
4. In the 'Add Permission Scheme' form, enter a name for the scheme, and a short description of the scheme. Select **Add**.
5. You will return to the 'Permission Schemes' page which now contains the newly added scheme.

## Adding users, groups, or roles to a permission scheme

1. Choose **Administration**  > **Issues**.
2. Select **Permission Schemes** to open the Permission Schemes page, which displays a list of all permission schemes in your Jira system and the projects that use each scheme.
3. Locate the permission scheme you would like to update, and select **Permissions** in the Operations column to view the scheme.
4. Select the '**Edit**' link for the permission you wish to add to, this displays the 'Grant permission' dialog.
5. Select who to add the selected permission to, and click the '**Grant**' button. The users/groups/roles will now be added to the selected permission. Note that **project roles** are useful for defining specific team members for each project. Referencing project roles (rather than users or groups) in your permissions can help you minimize the number of permission schemes in your system.
6. Repeat the last 2 steps until all required users/groups/roles have been added to the permissions.

## Deleting users, groups, or roles from a permission scheme

1. Choose **Administration**  > **Issues**.
2. Select **Permission Schemes** to open the Permission Schemes page, which displays a list of all permission schemes in your Jira system and the projects that use each scheme.

3. Locate the permission scheme of interest and click its name to show the list of 'Project Permissions' ([above](#)).
4. Click the **Remove** link for the permission you wish to remove the users, groups, or roles from.
5. Select the users, groups, or roles you wish to remove, and click the **Remove** button.

### Associating a permission scheme with a project

1. Choose **Administration** (⚙️) > **Projects**, and select the relevant project.
2. Select the project of interest to open the **Project Summary** administration page for that project. See [Defining a project](#) for more information.
3. On the lower right, in the **Permissions** section, click the name of the current scheme (e.g. 'Default Permission Scheme') to display the details of the project's current permission scheme.
4. Click the '**Actions**' dropdown menu and choose '**Use a different scheme**'.
5. On the 'Associate Permission Scheme to Project' page, which lists all available permission schemes, select the permission scheme you want to associate with the project.
6. Click the '**Associate**' button to associate the project with the permission scheme.

### Deleting a permission scheme

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Permission Schemes** to open the Permission Schemes page, which displays a list of all permission schemes in your Jira system and the projects that use each scheme.
3. Click the **Delete** link (in the **Operations** column) for the scheme that you want to delete.
4. A confirmation screen will appear. To delete click **Delete** otherwise click **Cancel**.
5. The scheme will be deleted and all associated projects will be automatically associated with the Default Permission Scheme. (Note that you cannot delete the Default Permission Scheme.)

### Copying a permission scheme

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Permission Schemes** to open the Permission Schemes page, which displays a list of all permission schemes in your Jira system and the projects that use each scheme.
3. Click the **Copy** link (in the **Operations** column) for the scheme that you want to copy.
4. A new scheme will be created with the same permissions and the same users/groups/roles assigned to them.

# Customizing Jira Service Management permissions

If you want to customize the permission scheme for your service project, make sure that you grant permissions to users by granting them:

- to the **Administrators** role for administrators
- to the **Service Desk Team** role for agents
- to the **Service Desk Customer - Portal Access** security type for customers.

If you grant permissions to groups or individual users instead of the roles and security type, some functionality in your service desk might be disabled.

## Default permissions

Default permissions are what we're using in the default permission scheme for service projects. Most of these permissions are optional, so you can choose which one to include in your customized permission scheme. For example, you might want to block users from adding attachments or setting the issue security although we've added these permissions to the default scheme.

To see the list of permissions, how they're assigned, and what they do, see [Permission overview](#).

## Mandatory permissions

Mandatory permissions are a subset of default permissions that affect some of the functionalities of your service project. In the table below, we've listed mandatory permissions for different project roles. Configuring them differently might result in errors and some of the functionalities not working properly. It's best to add these mandatory permissions and then decide which of the default permissions you need on top of that.

 If you do run into problems with mandatory permissions, see [Resolving Jira Service Management permission errors](#).

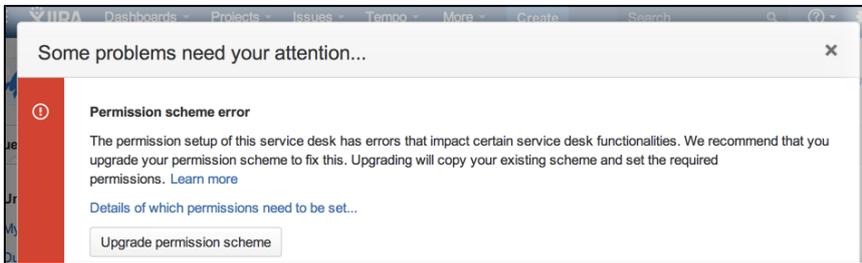
Project role	Mandatory permissions
<b>Administrators</b>	<p>This project role must have the <b>Administer Projects</b> project permission in order to set up and administer a service desk. This permission allows users to manage service desk functionality like creating new request types, setting up new queues, creating SLAs, and generating reports.</p> <p>This project role also must have all the permissions granted to the other users of the service desk in order to see all the functionality they'll be using.</p>
<b>Service Desk Team</b>	<ul style="list-style-type: none"><li>• Create Issues (This permission gives users the ability to create issues in a Customer Portal.)</li><li>• Browse Projects (This permission gives users read-only access to the <b>Reports, People</b>, and <b>SLA</b> tabs in a service desk project, as well as access to the project's Customer Portal. Users can also see the <b>Queue</b> tab and work on issues from within the queues.)</li><li>• Edit Issues</li><li>• Schedule Issues</li><li>• Add Comments</li><li>• Create Attachments</li></ul>

<b>Service Desk Customers</b>	<p>The permissions for customers must be granted to the <b>Service Desk Customer - Portal Access</b> security type, not the <b>Service Desk Customers</b> role. The Portal Access security type lets customers access the customer portal, but not Jira. The security type reads the role to determine who are customers.</p> <p>Mandatory permissions:</p> <ul style="list-style-type: none"><li>• Create Issues (Customers can create requests and view requests they have submitted via the customer portal)</li><li>• Browse Projects (Customers can access the project in the customer portal)</li><li>• Add Comments (Customers can comment on their requests.)</li></ul> <p>Recommended permissions:</p> <ul style="list-style-type: none"><li>• Create Attachments (Customers can add attachments to their requests)</li><li>• Assign Issue (This permission is mandatory for the <b>Assignee</b> field to work. The <b>Assignee</b> field is an optional hidden field and it automatically channels issues to certain team members.)</li></ul> <p><b>Note:</b> Jira will display a warning when you remove any of the permissions from the default scheme. In most cases, it will be a non-critical error (yellow), which doesn't affect how Service Desk works. If you removed the permissions on purpose, you can safely dismiss this warning, and it won't be shown again. You can read more about it in <a href="#">Resolving</a>.</p> <p>In addition, if the service desk project uses an issue security scheme, make sure that it is configured so that service desk users can view issues. Otherwise, customers might be able to create issues but not view them after they've been created. See <a href="#">Configuring issue-level security</a>.</p>
-------------------------------	--

# Resolving Jira Service Management permission errors

When you create a service project, it uses a permission scheme called *Jira Service Management Permission Scheme for %ProjectKey%*. If you change this permission scheme, then Jira Service Management might display a permission error similar to the following:

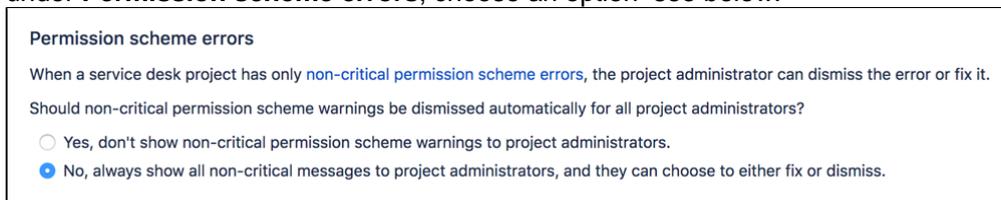
- [What are permission errors?](#)
- [How do I fix permission errors?](#)
- [Critical permission errors](#)



## What are permission errors?

Jira Service Management considers the differences between your permission scheme and the standard Jira Service Management permission scheme as errors in the following two categories:

- **Critical errors (red):** Break core service desk functionality, such as adding agents or allowing customers to log in to the portal. Jira Service Management displays a warning until you fix major errors. For a complete list of major errors, see [this table](#).
- **Non-critical errors (yellow):** Differ from the standard permission scheme, but don't impact how Jira Service Management works. Jira administrators can choose whether non-critical permission scheme warnings are dismissed automatically, or if they are always shown to project administrators, so that they can either fix or dismiss them.
  - Go to **Administration** (⚙️) > **Applications** > **Jira Service Management Configuration**, and under **Permission scheme errors**, choose an option see below:



## How do I fix permission errors?

To fix permission errors, you can change the permission scheme yourself, or click the **Fix permissions** button in the error message to have Jira Service Management fix the errors for you. When you click **Fix permissions**, Jira Service Management corrects the critical and non-critical errors in your permission scheme by doing the following:

1. Disassociates your permission scheme with the service desk project.
2. Creates a copy of your permission scheme called *%Your permission scheme%1* and associates the scheme with the project.
3. Fixes the errors by:
  - a. Granting standard permissions to the **Administrators** and **Service Desk Team** roles, and the **Service Desk Customer - Portal Access** security type.
  - b. Removing the **Service Desk Customers** role from all the permissions assigned.

The following table describes how Jira Service Management might fix a permission scheme:

<p><b>Custom permission scheme</b>  <b>Jira Service Management</b>  <b>Permission Scheme for</b>  <b>Project OA</b></p>	<p><b>Fixed permission scheme</b> <b>Jira Service Management</b> <b>Permission</b>  <b>Scheme for Project OA 1</b></p>
---	--

<p>The following permissions are set up differently from the standard permission scheme:</p> <ul style="list-style-type: none"> <li>• User <b>John Smith</b> has the <b>Browse Projects</b> permission. This is a minor error.</li> <li>• The <b>Service Desk Customers</b> role has the <b>Create Issues</b> permission. This is a major error.</li> <li>• The <b>Service Desk Customer - Portal Access</b> security type does not have the <b>Create Issues</b> permission. This is the major error.</li> </ul>	<p>After you click <b>Fix permissions</b>, the 'Jira Service Management Permission scheme for Project OA' permission scheme is dissociated with the project, and a new permission scheme called 'Jira Service Management Permission scheme for Project OA 1' will be applied to your service desk.</p> <ul style="list-style-type: none"> <li>• User <b>John Smith</b> will still have the <b>Browse Projects</b> permission.</li> <li>• The <b>Service Desk Customers</b> role is removed from the <b>Create Issues</b> permission.</li> <li>• The <b>Service Desk Customer - Portal Access</b> security type will be granted the <b>Create Issues</b> permission.</li> </ul>
---	--

## Critical permission errors

Critical permission errors break core service desk functionality. Jira Service Desk displays a warning until you fix them.

Error	Explanation
<p>The <b>Administrators</b> role does not have the following required permissions:</p> <ul style="list-style-type: none"> <li>• Browse Projects</li> <li>• Administer Projects</li> <li>• Edit Issues</li> </ul>	<ul style="list-style-type: none"> <li>• No <b>Browse Projects</b> permission = Administrators cannot access the service desk.</li> <li>• No <b>Administer Projects</b> permission = Administrators cannot modify settings of the service desk.</li> <li>• No <b>Edit Issues</b> permission = Administrators cannot edit issues.</li> </ul>
<p>The <b>Service Desk Customer - Portal Access</b> security type does not have the following required permissions:</p> <ul style="list-style-type: none"> <li>• Browse Projects</li> <li>• Create Issues</li> <li>• Add Comments</li> </ul>	<ul style="list-style-type: none"> <li>• No <b>Browse Projects</b> permission = Customers cannot access the Customer Portal of the service desk, that is they cannot log in.</li> <li>• No <b>Create Issues</b> permission = Customers cannot create requests on the Customer Portal.</li> <li>• No <b>Add Comments</b> permission = Customers cannot add comments to their requests.</li> </ul>
<p>The <b>Service Desk Customers</b> role is granted any permission directly.</p>	<p>Granting permissions to this role gives customers access to Jira functions. Customers should only have access to a Customer Portal and permissions should be granted to the <b>Service Desk Customer - Portal Access</b> security type.</p> <p>As a result, administrators will not be able to add any customers to the service desk. Open service desks will become restricted. Public signup will be disabled.</p>
<p>The <b>Service Desk Team</b> role does not have the following required permissions:</p> <ul style="list-style-type: none"> <li>• Browse Projects</li> <li>• Edit Issues</li> </ul>	<ul style="list-style-type: none"> <li>• No <b>Browse Projects</b> permission = Agents cannot see the service desk.</li> <li>• No <b>Edit Issues</b> permission = Agents cannot edit issues.</li> </ul>

<p>The <b>Service Desk Team</b> role is granted the Administer Projects permission.</p>	<p>Granting the Administer Projects permission to your agents means that all agents become administrators for your service desk.</p> <p>This is a severe security issue. Jira Service Management will disable the functionality of agent management. As a result, administrators will not be able to add any agents.</p>
<p>The <b>Anyone</b> group is granted the Browse Projects permission.</p>	<p>Granting the Browse Project permission to the Anyone group means that anyone can access the project and view all the issues in it.</p>

# Using Manage Sprints permission for advanced cases

The 'Manage Sprints' permission (only available to Jira Software users) is a [project permission](#) that allows users to perform the following sprint-related actions:

- Creating sprints (current and future ones)
- Starting sprints (current and future ones)
- Completing sprints
- Reopening sprints
- Reordering future sprints
- Deleting future sprints
- Editing sprint information (sprint name, dates, goals)
- Moving the sprint footer

## Caveats of the 'Manage Sprints' permission

With this permission, the board's filter query determines the projects that users need to have permission on. Also, permissions are now checked against the filter query of the board from which the sprint originates, not just against the issues within the sprint.

### When boards have complex filter queries

A filter query is considered complex when Jira Software can't determine which projects will be returned by the query. When this happens, Jira Software will require users to have the 'Manage Sprints' permission for all projects in the instance essentially, you'll need to manually set users to have this permission for all projects.

To handle this better, consider using [Jira project roles](#) for the 'Manage Sprints' permission. While project roles are defined at the instance level, they are applied at the project level. Thus, project level permissions can be given to [members of a project role](#), as well as groups, individual users, or through other means of designating a user. In essence, project roles enable you to associate users with particular functions for specific projects.

For example, you can consider doing the following:

1. [Create](#) a new [project role](#) called **Sprint Manager**.
2. In the corresponding permission scheme, assign the 'Manage Sprints' permission to the **Sprint Manager** project role.
3. [Associate](#) the permission scheme with the corresponding projects in your instance.
4. [Add](#) the appropriate users to the **Sprint Manager** project role.

Completing these steps will make sure that the appropriate users have the **Sprint Manager** project role in the corresponding projects and since the 'Manage Sprints' permission is assigned to the **Sprint Manager** project role, then these users can perform sprint-related actions.

The following table lists some examples of complex filter queries, and suggestions on simplifying such queries.

Complex filter query	Why the query is complex	How to simplify the query
assignee = someone	These queries return global context results because the results could potentially come from any project in the instance.	Add the project clause into the queries. This will reduce the number of projects Jira Software will check permissions on.
project = TIS OR issuetype = Bug		
project = TIS OR (issuetype = Bug AND assignee = someone)		

<pre>(project = TIS OR assignee = A)  AND  (project = PMO OR assignee = B)</pre>	<p>Jira Software will evaluate this query as:</p> <pre>(project = TIS AND assignee = B)  OR  (project = TIS AND project = PMO)  OR  (assignee = A AND project = PMO)  OR  (assignee = A AND assignee = B)</pre> <p>The <b>red parts</b> of the query won't return any results, which makes the query complex. Since the query returns undefined results, the 'Manage Sprints' permission will then be required for all projects in the instance.</p>	<p>Rewrite the query as:</p> <pre>(project = TIS AND assignee = B)  OR  (project = PMO AND assignee = B)</pre> <p>With this query, users will be required to have the 'Manage Sprints' permission on <i>only two projects</i>.</p>
--	--	--

In summary, we recommend that queries contain **OR** clauses in which **AND** clauses can be sub-clauses, and not the other way around.

Simply put, make sure:

- your **OR** clauses are outside the brackets, and
- your **AND** clauses sub-clauses are inside the brackets.

Recommended query format: <clause> OR (<clause> AND <clause>) OR <clause> OR (<clause> AND <clause>)

Complex query format: <clause> AND (<clause> OR <clause>) AND (<clause> OR <clause>)

### When boards contain sprints from other boards

With the 'Manage Sprints' permission, permissions are now checked against the filter query of the **origin board** the board from which the sprint is created not just against the issues within the sprint.

Depending on the filter query being used, your board might display sprints from other boards. For example, you have the **TIS board** and it's displaying Sprint 3, which was created in another **board** the **PMO board**. In this case, the **PMO board** is the **origin board** of Sprint 3.

If you're in the **TIS board** and you're closing Sprint 3, the following items are checked:

1. Jira Software checks if you have the 'Manage Sprints' permission for the projects in the **origin PMO board**.
2. If you have permissions, the sprint will be closed. If Sprint 3 has any incomplete issues, Jira Software will offer destination options, allowing you to move the incomplete issue to either the Backlog or a future sprint of the **TIS board**, e.g. Sprint 4.
3. If you choose to move the incomplete issue to Sprint 4, the issue is moved to Sprint 4 of the **TIS board**.
4. If Sprint 4 also exists in the **PMO board**, then the incomplete issue will appear in Sprint 4 of the **PMO board**.  
However, if Sprint 4 doesn't exist in the **PMO board**, the incomplete issue will be moved to the Backlog of the **PMO board**.

# Sprint permissions and defined processes

If processes in your organization are strictly defined then most likely people of different roles perform different sprint management operations. This article is to help you set up a proper permission scheme for working with sprints.

## 1. Select users who should manage sprints

The **Manage Sprints** permission, which gives you a complete set of rights for your sprint, should only be granted to users who need to fully manage the sprint.

With this permission, user can:

- Create sprints,
- Edit sprint properties, such as name, goal and dates,
- Delete sprints,
- Start/Complete sprints.

To set up permissions for users who handle sprint management (also from JiraAlign or Portfolio for Jira), go to **Project settings > Permissions** and modify the permission scheme by removing the **Manage Sprints** permission from everyone, besides the users who are responsible for creating and scheduling sprints.

This change should also be done for each project using the Scrum boards. It means, that if you have multiple permission schemes for different projects, you should make this change for each permission scheme.

Other roles who are not supposed to have such control over the sprint should only be granted more limited permissions.

## 2. Select the users who need more specific sprint permissions

A role that might need some but not all sprint permissions is the scrum master. This role possibly needs 2 permissions - **Start/Complete Sprints** and **Edit Sprint**.

A user with the **Start/Complete Sprints** permission can start sprints and end them when the sprint dates are set. This permission doesn't allow changing any sprint properties, such as the name, goal, and dates. You can only change the sprint properties to Active or Completed.

With the **Edit Sprints** permission users can change the sprint *name and goal*. However, they can't change sprint dates. This helps preserve the original sprint plan created by the RTE.

The members of the development team might also need some granular sprint permissions. For example, some of them might be granted the permission to **start/complete sprint** so, if need be, that they can stand in for the scrum master.

## 3. Set the permissions

To set these permissions, click a specific project and then go to **Project settings Permissions** and modify the permission scheme in a way described below:

1. Go to **Project settings > Permissions**.

The screenshot shows the Jira interface for a Scrum Project. The top navigation bar includes 'Jira', 'Dashboards', 'Projects', 'Issues', 'Boards', 'DbConsole', and a 'Create' button. The left sidebar contains navigation options for the project, including 'SP board', 'Backlog', 'Active sprints', 'Releases', 'Reports', 'Issues', and 'Components'. Below these are 'PROJECT SHORTCUTS' and an 'Add link' button. The main content area is titled 'Project settings' and lists various configuration options: Summary, Details, Re-index project, Delete project, Issue types (Bug, Epic, Story, Sub-task, Task), Workflows, Screens, Fields, Priorities, Versions, Components, Users and roles, **Permissions** (highlighted), Issue Security, Notifications, Project links, and Hipchat integration. The right sidebar shows a list of permissions, including 'Project Permissions', 'Administer', 'Browse Project', 'Edit Sprints', 'Manage Sprints', and 'View Development Information'.

2. Click **Actions > Edit permissions**.
3. Remove the **Edit Sprints** permission from everyone, besides the Scrum Masters.
4. Remove the **Start/Complete Sprints** permission from everyone, besides the Scrum Masters and Team leads.

5. Consider giving a **Start/Complete Sprints** permission to one of the team members, so that this person can stand for Scrum Master and Team Lead in case they are both unavailable.
6. Introduce these changes for each permission scheme your projects belong to. To see a list of projects assigned to this permission scheme click the **Used by projects** link under the permission scheme name.

The screenshot shows the Jira Administration interface. The top navigation bar includes 'Jira', 'Dashboards', 'Projects', 'Issues', 'Boards', 'DbConsole', and a 'Create' button. The main header is 'Administration' with a search bar 'Search Jira admin'. Below this is a secondary navigation bar with 'Applications', 'Projects', 'Issues' (highlighted), 'Manage apps', 'User management', and 'Latest upgrade report'.

The left sidebar contains a list of categories and their sub-items:

- ISSUE TYPES
  - Issue types
  - Issue type schemes
  - Sub-tasks
- BROWSE AND EXPORT
  - Archived issues
- WORKFLOWS
  - Workflows
  - Workflow schemes
- SCREENS
  - Screens
  - Screen schemes
  - Issue type screen schemes
- FIELDS
  - Custom fields
  - Field configurations
  - Field configuration schemes
- PRIORITIES
  - Priorities
  - Priority schemes
- ISSUE FEATURES
  - Time tracking

The main content area is titled 'Permission schemes' and shows the 'Default software scheme'. A red box highlights the 'USED BY 3 PROJECTS' link, which opens a dropdown menu listing the projects:

- Sample Kanban Project
- Scrum Project
- qq

Below the projects list, several permission categories are visible:

- Administer Projects**: Ability to administer a project in Jira. Includes a checked checkbox for 'Extended project administration' with the description 'Grant extended project administration permissions.'
- Browse Project Archive**: Ability to browse archived issues from a specific project.
- Browse Projects**: Ability to browse projects and the issues within them.
- Edit Sprints**: Ability to edit sprint name and goal.
- Manage Sprints**: Ability to manage sprints.
- Start/Complete Sprints**: Ability to start and complete sprints.
- View Development Tools**: Allows users in a software project to view development-related info and build information.
- View Read-Only Workflow**: Users with this permission may view a read-only version of a workflow.

 Note that the **Manage sprint** permission does not affect issue permissions by, for example, allowing for editing issue fields. It's only for managing the sprint entity.

# Managing project roles

Project roles are a flexible way to associate users and/or groups with particular projects. Project roles also allow for delegated administration:

- Jira administrators define project roles that is, all projects have the same project roles available to them.
- Project administrators [assign members](#) to project roles specifically for their project(s).  
*A project administrator is someone who has the project-specific 'Administrator Project' permission, but not necessarily the global 'Jira Administrator' permission.*

Project roles can be used in:

- [permission schemes](#)
- [email notification schemes](#)
- [issue security levels](#)
- comment visibility
- [workflow conditions](#)

Project roles can also be given access to:

- issue filters
- dashboards

Project roles are somewhat similar to groups, the main difference being that group membership is global whereas project role membership is project-specific. Additionally, group membership can only be altered by Jira administrators, whereas project role membership can be altered by project administrators. Every project has a project lead and every project component has a component lead. These individual roles can be used in schemes, issues and workflows, just like project roles. You assign project/component leads when [defining projects](#) or [managing components](#) respectively.

 For all of the following procedures, you must be logged in as a user with the **Jira Administrators** global permission.

## On this page:

- [Using project roles](#)
- [Default project roles](#)
- [Viewing project roles](#)
- [Adding a project role](#)
- [Deleting a project role](#)
- [Editing a project role](#)
- [Assigning members to a project role](#)
- [Specifying 'default members' for a project role](#)

## Using project roles

Project roles enable you to associate users with particular functions. For example, if your organization requires all software development issues to be tested by a Quality Assurance person before being closed, you could do the following:

1. [Create](#) a project role called **Quality Assurance**.
2. [Create](#) a permission scheme called **Software Development**, in which you assign the 'Close Issue' permission to the **Quality Assurance** project role.
3. [Associate](#) the **Software Development** permission scheme with all software development projects.
4. For each software development project, [add](#) the appropriate Quality Assurance people to the **Quality Assurance** project role.

## Default project roles

When you install Jira applications, the Administrators role is automatically created, along with project roles specific to each application. You can [create](#), [edit](#), and [delete](#) project roles according to your organization's requirements.

## Viewing project roles

1. In the upper-right corner of the screen, select **Administration**  **System**.

2. Select **Project roles** to open the Project Role Browser page.
3. You will then see the Project Role Browser, which contains a list of all the project roles in your Jira system.
4. To see where a project role is used, click the **View Usage** link. This will display a list of the project role's associated [permission schemes](#), [email notification schemes](#), [issue security levels](#), and [workflow conditions](#).
5. Click any of the **View** links on the 'View Usage for Project Role' screen to see which users/groups are associated with a project role for a particular project.

## Adding a project role

To define a new project role, enter its Name and a Description in the 'Add Project Role' form in the [project role browser](#) (see 'Viewing Project Roles' above), and click the **Add Project Role** button. Note that project role names must be unique.

1. Click on Manage Default Members in the **Operations** column for the newly created Project Role.
2. Click **Edit** under Default Users.
3. Select the User Picker icon to the right of the *Add user(s) to project role* field.
4. Click the Select button at the bottom of this dialog when you are finished adding users and then click the Add button. You now see a list of users on the right that are now included in this Project Role.

Once a new project role is created, it is available to all projects. Project administrators can then assign members to the project role for their project (see [Managing project role membership](#)).

## Deleting a project role

To delete a project role, locate the project role in the [project role browser](#) (see 'Viewing Project Roles' above), and click the **Delete** link. The confirmation screen that follows lists any [permission schemes](#), [email notification schemes](#), [issue security levels](#), and [workflow conditions](#) that use the project role.

Note that deleting a project role will remove any assigned users and groups from that project role, for all projects. Be aware of the impact this may have; for example, if the project role membership was the sole conveyor of a permission for a user, then the user will no longer have that permission.

 If a project role has been used to specify who can view a comment, deleting the project role will mean that no one can see that comment any more.

## Editing a project role

To edit the **Name** and **Description** of a project role, locate the project role in the [project role browser](#) (see 'Viewing Project Roles' above), and click the **Edit** link.

## Assigning members to a project role

A project role's members are assigned on a project-specific basis. To assign users/groups to a project role for a particular project, please see [Managing project role membership](#).

To see/edit *all* the project roles to which a particular user belongs, for all projects, click the **Project Roles** link in the user browser.

## Specifying 'default members' for a project role

The default members for a project role are users and groups that are initially assigned to the project role for all newly created projects. The actual membership for any particular project can then be [modified](#) by the project administrator.

The default members consist of the **Default Users** plus the **Default Groups** shown in the [project role browser](#) (see 'Viewing Project Roles' above).

To add to the **Default Users** or the **Default Groups** for a project role, click the corresponding **'Edit'** link.

For example, if a user called Susie needs to have administration permissions for all newly created projects, you could add her to the **Default Users** for the 'Administrator' project role as follows:

1. Open the [project role browser](#).
2. Click the **Manage Default Members** link.
3. Click the **Edit** link in the **Administrators** column (next to 'None selected').
4. In the 'Assign Default Users to Project Role' screen, click the **User Picker** icon.
5. Locate Susie in the 'User Picker' popup window, then click the **Select** button.
6. In the 'Assign Default Users to Project Role' screen, click the **Add** button.

 Changing a project role's default members does not affect the actual project role members for projects already created.

# Managing project role membership

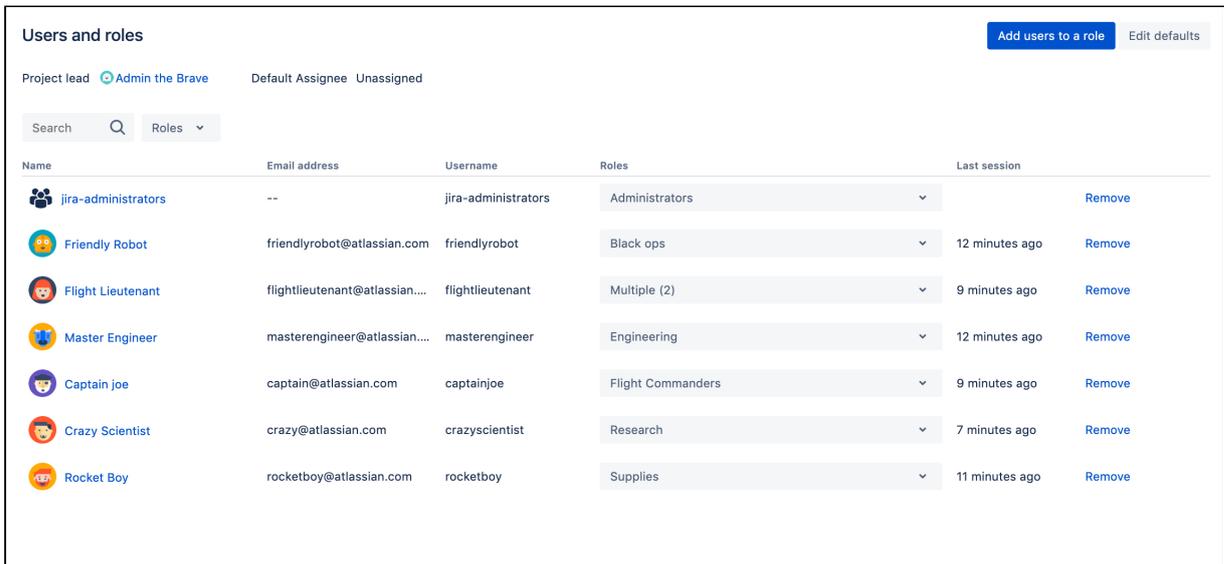
A Jira application project role is a flexible way to associate users and/or groups with a particular project. Unlike groups, which have the same membership throughout Jira applications, project roles have specific members for each project. Users may play different roles in different projects.

This page contains instructions for managing membership of *existing project roles*. For information on creating and using project roles, see [Managing project roles](#).

 For all of the following procedures, you must be logged in to Jira as a [project administrator](#).

## Viewing project role members

1. Choose **Administration**  > **Projects**, and select the relevant project.
2. Choose **Users and roles** in the left menu to view and manage project role membership.



Name	Email address	Username	Roles	Last session
 jira-administrators	--	jira-administrators	Administrators	
 Friendly Robot	friendlyrobot@atlassian.com	friendlyrobot	Black ops	12 minutes ago
 Flight Lieutenant	flightlieutenant@atlassian....	flightlieutenant	Multiple (2)	9 minutes ago
 Master Engineer	masterengineer@atlassian....	masterengineer	Engineering	12 minutes ago
 Captain joe	captain@atlassian.com	captainjoe	Flight Commanders	9 minutes ago
 Crazy Scientist	crazy@atlassian.com	crazyscientist	Research	7 minutes ago
 Rocket Boy	rocketboy@atlassian.com	rocketboy	Supplies	11 minutes ago

## Assigning a user or group to a project role

1. Open the **Users and roles** page as described [above](#).
2. Select **Add users to a role** from the top right corner.
3. Search for the user or group you wish to add, and select the project role you wish to add them to.

Note that the **Browse Users global permission** is required to search for existing users or groups at this step. If you do not have this permission, you will need to specify the exact name or email address.

4. Select **Add**.

After the user or group has been added to the list, you can quickly change their roles by using the drop-down menu in the *Roles* column. Just open the drop-down and select or clear roles assigned to this user.



- Because group membership can only be edited by users with the **Jira Administrator global permission**, project administrators may therefore prefer to assign users, rather than groups, to their project roles.
- A project role does not need to have any users or groups assigned to it, although project administrators should be careful with this. Depending on how a project role is used (e.g. if the project's [permission scheme](#) is using project roles), it is possible that not having anyone in a particular project role could make some project activities unavailable.

# Allowing anonymous access to your project

 For all of the following procedures, you must be logged in as a user with the **Jira project administrator** permissions. For details, see [Permissions overview](#).

Jira applications can be configured to allow users to create issues without having logged in. To do this, you need to allow anonymous users to browse, search, and create issues in that project.

Anonymous users will have access equivalent to Jira Core users. This means that they can view issues and work in any type of project, but they won't see application-specific features, like agile boards, which are Jira Software-specific features. See [Jira applications and project types overview](#) for more information.

## Allow anonymous users to browse and search for issues in the project

Add the Anyone on the web group to the **Browse Project** permission in the permission scheme for the project:

1. Go to **Administration**  > **Issues**.
2. Select **Permission Schemes** to open the [Permission Schemes](#) page for your project. This page displays a list of all permission schemes in your Jira system and the projects that use each scheme.
3. Find the permission scheme you want to update, then select **Permissions** in the **Actions** column to view the scheme.
4. Select the **Edit** link for the **Browse Projects** permission.
5. In the **Grant permission** dialog, select **Granted to > Group**, and choose **Anyone on the web** from the dropdown list.
6. Select **Grant** to save your changes.

## Allow anonymous users to create issues in the project

Add the Anyone on the web group to the **Create Issue** permission in the permission scheme for the project:

1. Go to **Administration**  > **Issues**.
2. Select **Permission Schemes** to open the [Permission Schemes](#) page for your project.
3. Find the permission scheme you would like to update, and then select **Permissions** in the **Actions** column to view the scheme.
4. Select the **Edit** link for the **Create Issues** permission.
5. In the **Grant permission** dialog, select **Granted to > Group**, and choose **Anyone on the web** from the dropdown list.
6. Select **Grant** to save your changes.
7. You also need to set the **Reporter** in the project's field configuration scheme to optional (see [Making a field required or optional](#) for instructions). After you update the field, any issue created by a user who is not logged in will display Anonymous for the reporter of the issue.

 Make sure that the anonymous user is able to complete and submit all required fields. For example, if you make the Due Date field required, the anonymous user will also need to have the Schedule Issue permission.

## Disable anonymous access to your project

Remove the Anyone on the web group from **Create Issue** and **Browse Projects** permissions:

1. Go to **Administration**  > **Issues**.
2. Select **Permission Schemes** to open the [Permission Schemes](#) page for your project.
3. Select the **Remove** link for permission you want to update, and remove the permission for the **Anyone on the web** group.

# Managing components

Components are sub-sections of a project. They are used to group issues within a project into smaller parts. You can set a [default assignee](#) for a component. This will override the project's default assignee, for issues in that component.

You need to have the project-specific **Administer Projects** permission or the **Jira Administrator** global permission to be able to:

- Add create a new component against which issues can be aligned
- Select a default assignee choose who is automatically assigned issues with that component assigned
- Edit change a components details
- Delete remove a component

Once a component has been created for a project, the 'Component' field becomes available for your issues. If you cannot see this field on your issue, your project may not have any components yet, or the field is hidden from view.

## On this page:

- [Managing a project's components](#)
- [Adding a new component](#)
- [Editing a component's details](#)
- [Searching for a component](#)
- [Deleting a component](#)
- [Archiving a component](#)
- [Restoring a component](#)

## Managing a project's components

1. Choose **Administration** (⚙️) > **Projects**, and select the relevant project.
2. Choose **Components** in the project sidebar. The **Components** page is displayed, showing a list of components and each component's details. From here you can manage the project's components as described below.

## Adding a new component

1. The Add Component form is located at the top of the 'Components' screen.
2. Enter the **Name** for the component. Optionally, enter a **Description**, and select a **Component Lead** and **Default Assignee** (see [options](#) below).
3. Click **Add**.

## Selecting a Default Assignee

You can optionally set a Default Assignee for a component. This will override the project's default assignee for issues in that component.

### Multiple components and selecting assignees

In the case of issues with multiple components, the assignee structure works top-down, that is from the least to the most specific. The most specific (Component Lead) takes precedence over others. For example, if an issue has multiple components and some of these components have the Unassigned option selected, some have a Project Lead specified, and some others - a Component Lead, then the Component lead will always take precedence as the Assignee.

Default assignee option	Description	Notes
Project Default	Issues matching this component will have the assignee set to the same default assignee as the parent project.	 If an issue has multiple components, and no assignee is selected for any of the components, we use the Project Default assignee.

Unassigned	The assignee of the issue will not be set on the creation of this issue.	This option will only be available if "Allow unassigned issues" is enabled in the <a href="#">General Configuration</a> .   Unassigned takes precedence over <i>Project Default</i> as the Assignee.
Project Lead	The assignee will be set to the project leader.	If the project leader is not permitted to be assigned to issues in the <a href="#">permission scheme</a> , this option will be disabled and will say "Project Lead is not allowed to be assigned issues".   Project Lead takes precedence over <i>Project Default</i> and <i>Unassigned</i> as the Assignee.
Component Lead	The assignee will be set to the component leader.	If the component leader is not permitted to be assigned to issues in the <a href="#">permission scheme</a> , this option will be disabled and will say "Component Lead is not allowed to be assigned issues". The Component Lead option will also not be available if the component does not have a lead assigned to the component. Instead under this option it will say "Component does not have a lead".   If an issue has multiple components, and at least one of them uses Component Lead, then the Component Lead takes precedence over <i>Project Lead</i> , <i>Project Default</i> , and <i>Unassigned</i> as the Assignee. If the components have different Component Leads, the default assignee will be set to the Component Lead of the component that is first alphabetically.

## Editing a component's details

1. On the 'Components' screen, open the menu in the **Actions** column for the component you want to edit, and select **Edit**.
2. Edit the component's **Name**, **Description**, **Lead**, and **Default Assignee** in the **Edit component** dialog.
3. Click the **Save** button to save your changes.

## Searching for a component

If you need to find a component in a long list, it's easiest to search for it. Start typing text into the search box that you know the component contains, and your list will automatically be filtered for you.

## Deleting a component

1. On the 'Components' screen, open the menu in the **Actions** column for the component you want to delete, and select **Delete**.
2. You will be prompted to associate any issues assigned to this component with another component if you wish.
3. Click the **Delete** button to delete the component.

## Archiving a component

You can archive the components that are no longer relevant so that they do not clutter your Jira instance. By archiving a component, you make it unavailable to be chosen and linked to issues but you still keep it as reference for reporting purposes. The component still appears on the components' list but is marked as Archived.

To archive a component:

1. Open your project, and click **Components** in the navigation bar on the left.
2. Find the component you want to archive, and select **More** (  ) > **Archive**.

The component will be immediately tagged as Archived.

The component remains on the components' list, marked as **Archived**. It disappears from the components dropdown on an issue page so that it cannot be selected and linked to a new issue. Archived components linked to issues show as read-only in the Components field.

Affects Version/s:	None
Component/s:	<input type="text" value="New component x"/> 
	Archived Component/s: Old component
Labels:	None  
Sprint:	Sample Sprint 2

If you use Jira Service Management as well, then your archived components will show as read-only in tickets. Additionally, you will not be able to select and link an archived component to the ticket.

### Restoring a component

All the component data remains in Jira, so you can restore it whenever it's needed again.

To restore a component:

1. Go to **Components** and select the component to restore. Tip: You can filter for the **Archived** status to find the component you need.
2. Next to the component to be restored click **More** (  ) > **Restore**.

The component will be restored to its original state and marked **Active**.

# Archiving a component

Archive the components that are no longer relevant so that they do not clutter your Jira instance. By archiving a component, you make it unavailable to be chosen and linked to issues but you still keep it as reference for reporting purposes. The component still appears on the components' list but is marked as Archived.

- [Before you begin](#)
- [Archiving a component](#)
  - [What happens to a component after you archive it?](#)
- [Restoring a component](#)

## Before you begin

- You must have the Jira Administrator or Jira System Administrator permission to archive or restore components.
- You can archive any component, and restore it later, if needed.

## Archiving a component

To archive a component:

1. Open your project, and click **Components** in the navigation bar on the left.
2. Find the component you want to archive, and select **More** (  )> **Archive**.

The component will be immediately tagged as Archived.

## What happens to a component after you archive it?

The component remains on the components' list, marked as **Archived**. It disappears from the components dropdown on an issue page so that it cannot be selected and linked to a new issue. Archived components linked to issues show as read-only in the Components field.

Affects Version/s:	None
Component/s:	<input type="text" value="New component x"/> 
	Archived Component/s: <b>Old component</b>
Labels:	None  
Sprint:	Sample Sprint 2

*Only applicable if Jira Service Management is installed and licensed.*

- Archived components show as read-only in tickets.
- You cannot select and link an archived component to the ticket.

## Restoring a component

All the component data remains in Jira, so you can restore it whenever it's needed again.

To restore a component:

1. Go to **Components** and select the component to restore. Tip: You can filter for the **Archived** status to find the component you need.
2. Next to the component to be restored click **More** (  )> **Restore**.

The component will be restored to its original state and marked **Active**.



# Managing versions

Versions are points-in-time for a project. They help you schedule and organize your releases. Once a version is created and issues are assigned to it, you can use several reports, e.g. the Change Log report, when managing the version. The Change Log report, in particular, gives you a review of the released version, and is driven by the 'Fix For Version' field on each issue.

Versions can be:

- **Added** create a new version against which issues can be aligned.
- **Released** mark a version as released. This makes some changes in some reports (e.g. Change Log report) and some issue fields' drop-downs. If you have integrated Jira applications with [Bamboo](#), you can also trigger builds when releasing a version.
- **Rescheduled** re-arrange the order of versions.
- **Archived** hide an old version from the Change Log reports, and in the Jira User Interface.
- **Merged** combine multiple versions into one.

## On this page:

- [Managing a project's versions](#)
- [Add a new version](#)
- [Add a start date](#)
- [Release a version](#)
- [Archive a version](#)
- [Merge multiple versions](#)
- [Edit a version's details](#)
- [Delete a version](#)
- [Reschedule a version](#)

 For all of the following procedures, you must be logged in to Jira as [a project administrator](#).

## Managing a project's versions

1. Choose **Administration**  > **Projects**, and select the relevant project.
2. Choose **Versions** in the left menu. The **Versions** page is displayed, showing a list of versions and each version's status. From here you can manage the project's versions as described on this page.

### Version status

Each version can have any of the following four statuses:

- **Released** a bundled package
- **Unreleased** an open package
- **Archived** a semi-transparent package
- **Overdue** the release date is highlighted

 The status affects where the version appears in drop-down lists for version-related issue fields ('Fix For Version' and 'Affects Version').

## Add a new version

1. The Add Version form is located at the top of the Versions screen.
2. Enter the name for the version. The name can be:
  - simple numeric, e.g. "2.1", or
  - complicated numeric, e.g. "2.1.3", or
  - a word, such as the project's internal code-name, e.g. "Memphis".
3. Optional details such as the version description (text not HTML), start date and release date (i.e. the planned release date for a version) can also be specified.
4. Click the **Add** button. You can drag the new version to a different position by hovering over the 'drag' icon  at the left of the version name.

## Add a start date

If specified, the **Start Date** is used by the Version Report. This gives you a more accurate report in cases where you might plan a version many weeks (or even months) in advance, but not actually commence work until closer to the release date.

## Release a version

**Before you begin:** If you have integrated Jira applications with [Atlassian's Bamboo](#), you can trigger a Bamboo build to run automatically when releasing a version in Jira. The version will only be released if the build is successful. For more information, see [Running a Bamboo build when releasing a version](#).

1. On the Versions screen, hover over the relevant version to display the cog icon, then select **Release** from the drop-down menu.
2. If there are any issues set with this version as their 'Fix For' version, Jira applications allow you to choose to change the 'Fix For' version if you wish. Otherwise, the operation will complete without modifying these issues.

To revert the release of a version, simply select **Unrelease** from the drop-down menu.

## Archive a version

1. On the Versions screen, hover over the relevant version to display the cog icon, then select **Archive** from the drop-down menu.
2. The version list indicates the version 'archived' status with a semi-transparent icon. The list of available operations is replaced with the 'Unarchive' operation. No further changes can be made to this version unless it is un-archived. Also it is not possible to remove any existing archived versions from an issue's affected and fix version fields or add any new archived versions.

To revert the archive of a version, simply select **Unarchive** from the drop-down menu.

## Merge multiple versions

Merging multiple versions allows you to move the issues from one or more versions to another version.

1. On the Versions screen, click the **Merge** link at the top right of the screen.
2. The 'Merge Versions' popup will be displayed. On this page are two select lists both listing all versions.  
In the 'Merging From Versions' select list, choose the version(s) whose issues you wish to move. Versions selected on this list will be removed from the system. All issues associated with these versions will be updated to reflect the new version selected in the 'Merge To Version' select list. It is only possible to select one version to merge to.
3. Click the **Merge** button. If you are shown a confirmation page, click **Merge** again to complete the operation.

## Edit a version's details

1. On the 'Versions' screen, hover over the relevant version to display the pencil icon.
2. This will allow you to edit the version's Name, Description and Release Date.
3. Click the **Update** button to save your changes.

## Delete a version

1. On the 'Versions' screen, hover over the relevant version to display the cog icon, then select **Delete** from the drop-down menu.
2. This will bring you to the 'Delete Version: <Version>' confirmation page. From here, you can specify the actions to be taken for issues associated with the version to be deleted. You can either associate these issues with another version, or simply remove references to the version to be deleted.

## Reschedule a version

Rescheduling a version changes its place in the order of versions.

- On the 'Versions' screen, click the  icon for the relevant version, and drag it to its new position in the version order.

# Creating release notes

Jira provides the functionality to create release notes for a specific version of a project. The release notes contain all issues within the specified project that are marked with a specific "Fix For" version. The release notes can also be generated in a number of formats (e.g. HTML, plain text, etc.) so as they can be included in various documents.

At present, two example format templates are provided - HTML and Text - using Velocity templates. Further format templates can be created and added to the system.

## Generating Release Notes

1. Select **Projects** in the navigation bar.
2. Select your project from the list, or select **View all projects** and navigate to your project.
3. Select **Versions** in the project sidebar (if you're using Jira Software, select **Releases** in the project sidebar).
4. Select the **Version** whose release notes you wish to generate by clicking on it.
5. Select **Release Notes**.
6. Click the '**Configure Release Notes**' link to configure the release notes. The 'Configure Release Notes' page will be displayed:
  - Select the required project version for which the release notes will be generated in the '**Please select version**' drop-down.
  - Select the required format of the release notes. HTML and plain text format templates are provided in the '**Please select style**' drop-down.
7. Selecting the '**Create**' button will generate the release notes using the specified template in the specified format. The release notes will be displayed on screen and can be copied and pasted to another application.

## Adding a New Format Template

1. Create a Velocity template similar in content to that of the examples provided `releasenotes-text.vma` and `releasenotes-html.vm`. Consult the [Jira API documentation](#) and the [Apache Velocity User Guide](#).
2. The title within the template should be modified along with the code within the text area. The other sections of the template do not need to be modified.
3. Add the new format template to the list of existing ones within the `jira-config.properties` file. For each new template format, corresponding entries must be added to the existing values of the following properties:
  - `jira.releasenotes.templatenames`
  - `jira.releasenotes.templates`

### Notes:

- a. Corresponding entries in both of these properties must be in the same order.
- b. If these properties do not exist in your `jira-config.properties` file, then:
  - i. For each of these properties, add the property's name,
  - ii. followed by an '=',
  - iii. followed by the content of the property's corresponding `<default-value/>` element copied from your Jira installation's `jpm.xml` file.
  - iv. Next, begin adding the corresponding entries for the new format template.

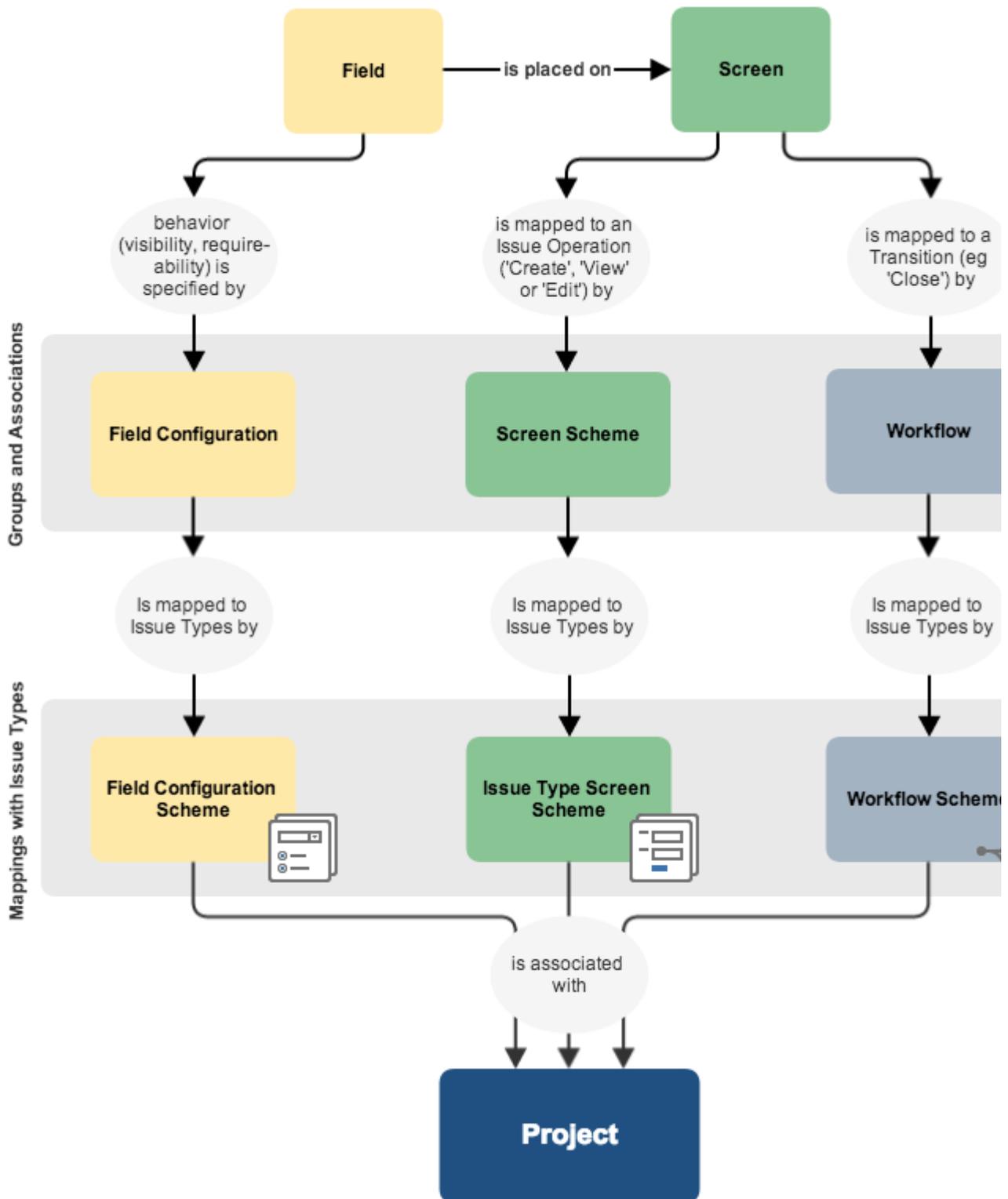
📌 See [Making changes to the jira-config.properties file](#) for more information.

4. The new format template is available for selection as a release note format template.

Also see the tutorial on [Creating a Custom Release Notes Template Containing Release Comments](#).

# Project screens, schemes and fields

Information for each issue is held in the fields that are associated with that issue. You can tailor these fields to suit your organization's needs. The diagram below is a representation of how these fields are associated with an issue, via screens and schemes. A screen is the user's view of an issue, and the screen is mapped to a specific issue operation (such as creating an issue, or editing an issue) via a screen scheme. The screen scheme is then mapped to an issue type via the issue type screen scheme. This configuration is associated with the project, and is applicable to all issues within the project.



Customizing the fields, screens and schemes allows you to unlock the full power of your Jira application, and ensure that your users are working efficiently and effectively. You can also set up notification schemes which will notify your users when their issues have been updated. The following pages in this section will help you to configure and customize Jira to suit your needs.

Adding a custom field	<a href="#">Learn more</a> about how custom fields work, and how you can add them to your issues so you make sure you get the information you need on each issue.
Configuring a custom field	<a href="#">Learn more</a> about how to modify each of the custom fields to change their name, description, default value, and more.
Specifying field behavior	<a href="#">Learn more</a> about how you can change how a field behaves, and when it displays, to make sure your users always see and record the information that's most important.
Defining a screen	<a href="#">Learn more</a> about how you can change what displays on each screen, and how to associate the screens with issues and issue operations.
Notification schemes	<a href="#">Learn more</a> about how you can create notification schemes that keep your users updated when there are updates on their issues.
Optimizing custom fields	<a href="#">Learn more</a> about how to find custom fields whose configuration is not optimal for your Jira instance, and improve them.

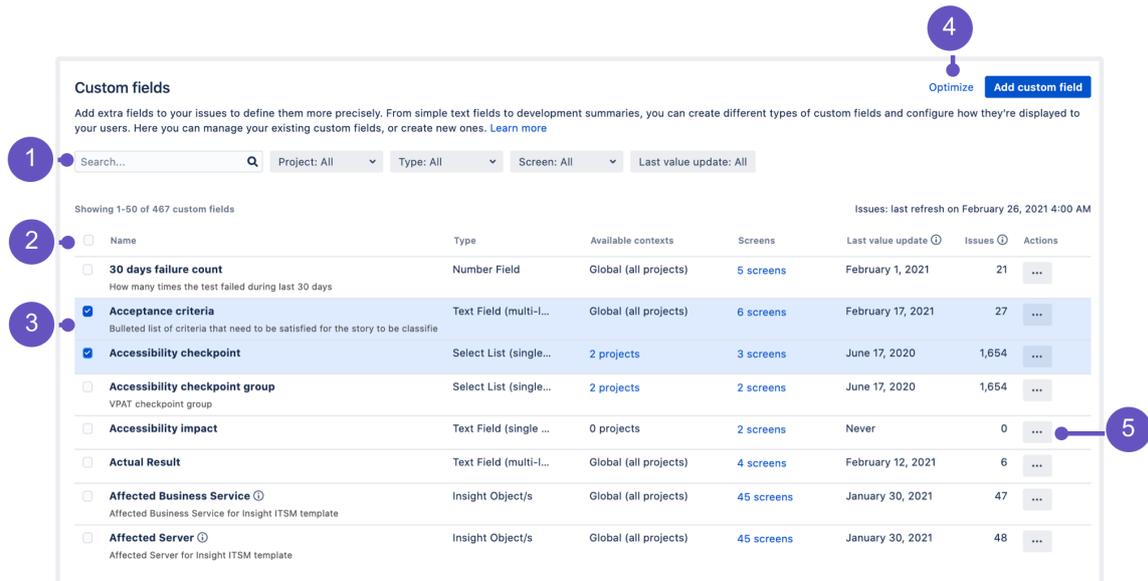
# Managing custom fields

Issues in Jira are made up of fields that store various pieces of information. Every issue comes with default system fields, like name or description, but you'll most likely need to add more fields to collect information that is very specific to your company. That's where custom fields come in; they can collect any info you need them to collect and be displayed in different configuration in your issues.

## Viewing custom fields

To view and manage your custom fields:

1. Go to **Administration > Issues**.
2. Click **Custom fields**.



1. **Search and filters:** As the list can get lengthy, here are some tools to help you find the right field.
2. **Columns:** All the columns to help you identify the field itself and the importance of it.
3. **Bulk delete:** Lets you select and bulk delete your fields, which is good for some spring cleaning.
4. **Optimize custom field contexts:** A way to improve the performance of your instance by narrowing down the range of your fields.
5. **Actions:** Configure your fields by adding contexts, screens, or translations.

## Identifying your fields

Every field is described by a set of columns to help you identify your fields and their importance within your Jira instance.

Column name	Description
<b>Name and Type</b>	Name, description, and type of a custom field. Make sure to describe your fields precisely, so they can be easily identified later on.  See <a href="#">Editing or deleting custom fields</a> .
<b>Available contexts</b>	Contexts are different configurations of the same field. For example, you can configure a field to use different values for different projects or issue types, or make it available only in some projects.  See <a href="#">Configuring custom field contexts</a> .

<b>Screens</b>	Screens are all these different forms that you fill in when creating issues, editing them, or transitioning them through workflows. Every custom field needs to be associated to some screens, or it won't be visible anywhere.
<b>Last value update (Data Center)</b>	Shows the last time a value for a field was added or updated in any issue, and should be used to identify custom fields that haven't been used for a long time, but still clutter your instance.  See <a href="#">Analyzing the usage of custom fields</a> .
<b>Issues (Data Center)</b>	Shows the number of issues that have a value defined for a field, which can be seen as issues actually needing it. This includes issues with default values and archived issues. Together with Last value update, this column should be used to identify unused fields.  See <a href="#">Analyzing the usage of custom fields</a> .

## Managing custom fields

What would you like to do with your custom fields?

- [Adding custom fields](#)
- [Configuring custom field contexts](#)
- [Editing or deleting custom fields](#)
- [Translating custom fields](#)
- [Analyzing the usage of custom fields](#)
- [Optimizing custom field contexts](#)

# Adding custom fields

You can create a custom field to collect information that isn't available in the default system fields. Before you start, make sure to read our tips for creating custom fields so you don't end up cluttering your Jira instance with too many fields.

**Custom fields are always optional fields.** This means that you can create a new custom field without requiring existing issues to be changed. The existing issues will contain no value for the new custom field, even if a default value is defined.

 For all of the following procedures, you must be logged in as a user with the **Jira Administrators** global permission.

## Tips for creating custom fields

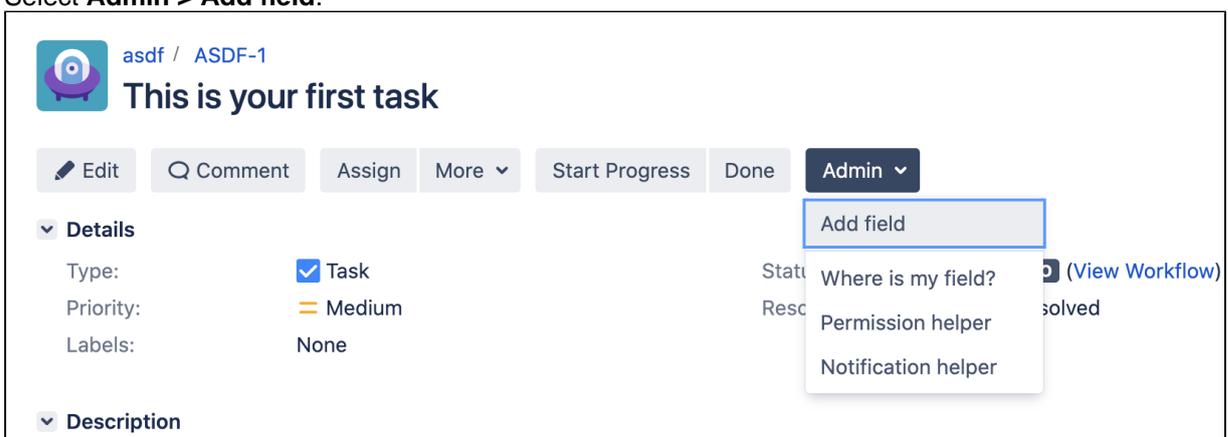
After creating thousands of custom fields, here are some best practices we'd like to share:

- **Limit the number of custom fields** Be careful how many custom fields you define in Jira. More than a thousand is a large number and may affect Jira's performance.
- **Combine field content** If you just want to make sure that someone remembers to enter some information, then consider a multi-line custom text field with a text template as a default value. You may also want to try using a "table grid" custom field which lets you enter data in a searchable table.
- **Don't duplicate names** Don't create new custom fields with the same name as other existing custom fields. Always check to see whether a custom field with the same name already exists before you create it. If you do, then choosing the correct field in JQL searches can become confusing for users. Also, don't create custom fields with the same name as the standard Jira fields. For example, having two "Status" fields is particularly confusing.
- **Make names as generic as possible** Give custom fields non-specific names that can be reused in other places later on. For example, instead of naming a field "Marketing Objective", name the field "Objective", and provide a description in the field configuration that states the Jira projects where that field is used.

## Add a field directly to an issue

To add a field to an issue:

1. Open an issue.
2. Select **Admin > Add field**.



## Add a field from the custom fields view

To add a field on the Custom fields page:

1. Go to **Administration > Issues**, and select **Custom fields**.
2. Select **Add custom field**.
3. In **Select a Field Type**, select **All** to make sure you can see all available field types.

4. Select a field type and click **Next**.
5. Configure the selection criteria for your field, as shown in the example for the Checkboxes field below:

### Configure 'Checkboxes' field

Name\*

Description

Options\*

Apples	✕
Oranges	✕
Pears	✕

**i** The **Field Name** will appear as the custom field's title in both entering and retrieving information on issues, whereas the **Field Description** is displayed beneath the data entry field when entering new issues and editing existing issues, but not when browsing issues.

6. Select **Next**.
7. Configure the context for your field and click **Next**.

Custom field contexts are different configurations of the same field. By configuring multiple contexts, you can use different default values and options for certain projects, or make the field be applicable only to chosen issue types or projects. It's generally a good idea performance-wise to limit a custom field to only the projects that will actually use it. [You can configure context later](#)

### Configure context

Select issue types\* All issue types × ▼

Enables the custom field for the selected issue types in the context specified above.

Choose context  **Apply to issues in selected projects** RECOMMENDED

Creates a field only in issues that belong to selected projects. This option has a low impact on the indexing time..

**Apply to all issues on your instance**

Creates a field for all future and existing issues in Jira. This option has a great impact on the indexing time.

Select projects\*   ▼

Previous
Create
Cancel

8. Click **Create**.

## Associating your field with a screen

After you create a custom field, you'll be moved to the Screens page. Here you can choose what screens your custom fields should be displayed on, for example Create Issue screen or Edit screen.

1. Have a look at the list of available screens and their projects.
2. Select a checkbox next to the desired screen, and click **Update**.

## Next steps

If you wish to change the context, see [Configuring a custom field](#). You can also find more custom fields from apps listed on the [Atlassian Marketplace](#) (e.g. the [Jira Toolkit](#)). To build your own custom field types, see the [tutorial](#) at the [Jira Developer Documentation](#).

### Do more with Jira

To help you add custom fields to Jira, use one of the apps available on the [Atlassian Marketplace](#):

- [Profields - Jira Project Custom Fields](#): Add, manage, and customize Jira project fields according to each project needs
- [Power Custom Fields](#): Use formulas, scripts, calculated fields, and custom messages inside a custom field
- [Elements Connect - external data fields](#): Create custom fields which can be populated with any datasource

# Configuring custom field contexts

Custom field contexts are different configurations of the same field. Instead of creating multiple custom fields with different default values for your projects, you can achieve the same results by using contexts.

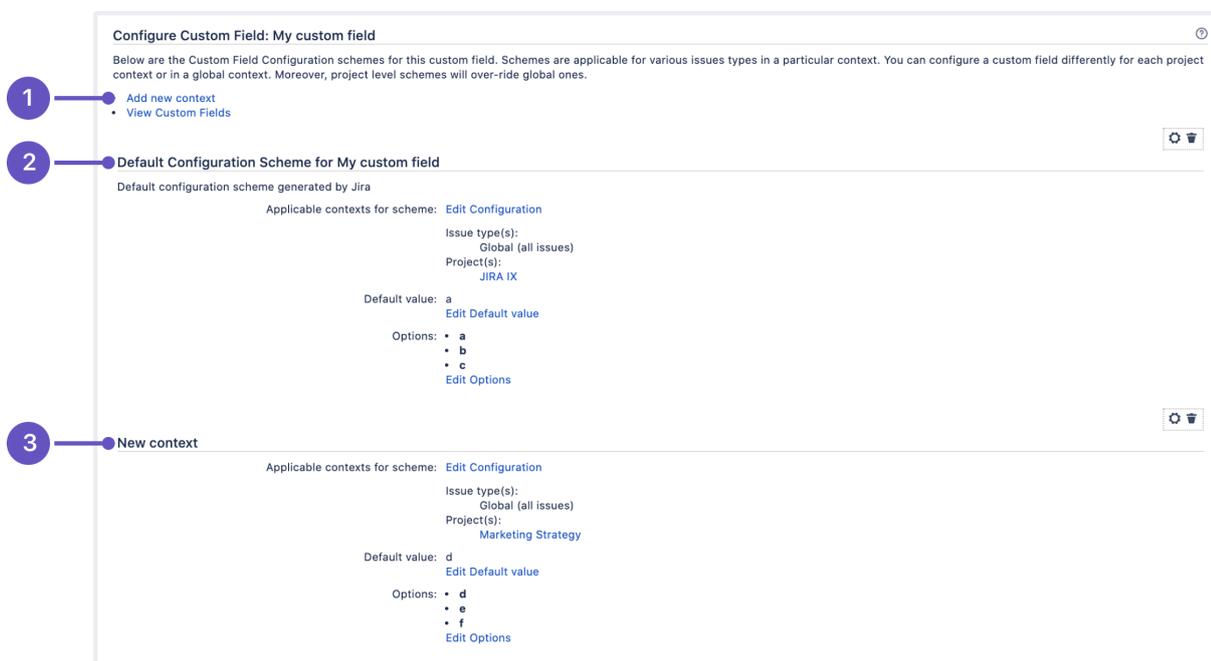
For every context, you can choose the following items:

- Default value (what is pre-filled when a custom field is displayed)
- Options (what users can choose from)
- The issue types and projects to which these default values and options apply

## Viewing and configuring contexts

To view existing contexts for you field and to add some new ones:

1. Go to **Administration > Issues**, and select **Custom fields**.
2. Find your custom field, and select **Actions ( ... ) > Configure**.



1. **Add new context:** Here you can add a new context, or go back to the list of custom fields.
2. **Default context:** Details of the default context, with options, default value, and projects chosen when the custom field was created.
3. **Another context:** A new context for this field, with different options, default value, and projects.

## Changing the default context

Each custom field has a context named *Default Configuration Scheme for*, which is created automatically when you add your custom field. It includes all the details you specified when creating your field.

- To change the details of this context, like name and description, and the issues and projects this context applies to, select **Edit configuration**.
- To change the default value, select **Edit default value**.
- To change the options, select **Edit options**. This will not be available for custom field types that don't use options.

## Adding a new context

To add a new context:

1. At the top of the page, select **Add new context**.
2. Choose a name for this context and a helpful description.

3. Select the issue types and projects this context should apply to.
4. Click **Add**.

After you add a new context, you can change its details, default values, and options:

- To change the details of this context, like name and description, and the issues and projects this context applies to, select **Edit configuration**.
- To change the default value, select **Edit default value**.
- To change the options, select **Edit options**. This will not be available for custom field types that don't use options.

## Adjusting values in existing issues after changing the custom field configuration

When you edit the values in your custom field configuration, you should manually update the values in existing issues. If you don't do that, then as soon as you edit an issue the custom field values will revert to null or the default option on the issue page.

- If the field is **optional**, the field will be set to 'None' on the Edit screen.
- If the field is **required**, the field will be set to the first available option, and not the default option, on the Edit screen.

To find and edit the issues:

1. Search for the issue or issues that use the custom field value you have changed.
2. Bulk edit the issues to set them to the correct value in the new field configuration (even if that value matches the default field config).

For detailed description and steps to reproduce to avoid data loss, see [Change context for an existing custom field without losing old data](#).

# Editing or deleting custom fields

You can edit or delete custom fields directly from the Custom fields page. For Jira Data Center, you can also delete multiple custom fields at once.

## Before you begin

- When you delete a custom field, its gone forever and this includes all data stored against this field.
- If a custom field is used on boards, filters, and workflows, deleting it might also affect these items.

## Edit a custom field

To edit a custom field:

1. Go to **Administration > System**, and select **Custom fields**.
2. Find your custom field, and select **Actions** (  ) > **Edit**.
3. You can update the following details:
  - The custom field **name**, which appears on issues.
  - The custom field **description**, which appears below the field on issues.
  - **Search templates**, which are responsible for indexing a custom field and making the field searchable in basic and advanced issue search. Custom fields come with a default search template, so you shouldn't need to change this setting.
4. Select **Update** to save your changes.

## Delete a custom field

To delete a custom field:

1. Go to **Administration > System**, and select **Custom fields**.
2. Find your custom field, and select **Actions** (  ) > **Delete**.

## Bulk delete custom fields

In Jira Data Center, you can delete multiple fields at once:

1. Go to **Administration > System**, and select **Custom fields**.
2. Use filters to find the custom fields you want to delete. Its also a good idea to sort by usage columns (Issues, Last value update) to find custom fields that arent used too often.

 Its also a good idea to sort by the usage data (Issues, Last value update) to find custom fields that arent used too often. For more info on analyzing your custom fields, see [Analyzing the usage of custom fields](#).

3. Select the check boxes to the left of your fields to select them, and then click **Delete**.

### Custom fields Optimize [Add custom field](#)

Add extra fields to your issues to define them more precisely. From simple text fields to development summaries, you can create different types of custom fields and configure how they're displayed to your users. Here you can manage your existing custom fields, or create new ones. [Learn more](#)

Search...  Project: All  Type: All  Screen: All  Last value update: All

Showing 1-50 of 475 custom fields Issues: last refresh on March 11, 2021 4:00 AM

4 custom fields selected

<input checked="" type="checkbox"/>	<b>30 days failure count</b> How many times the test failed during last 30 days	Number Field	Global (all proje...	5 screens	February 1, 2021	21	<input style="background-color: #ccc; border: 1px solid #ccc; padding: 2px 5px;" type="button" value="..."/>
<input checked="" type="checkbox"/>	<b>Acceptance criteria</b> Bulleted list of criteria that need to be satisfied for the story to be cla	Text Field (multi...	Global (all proje...	6 screens	March 4, 2021	28	<input style="background-color: #ccc; border: 1px solid #ccc; padding: 2px 5px;" type="button" value="..."/>
<input checked="" type="checkbox"/>	<b>Accessibility checkpoint</b>	Select List (sing...	2 projects	3 screens	June 17, 2020	1,654	<input style="background-color: #ccc; border: 1px solid #ccc; padding: 2px 5px;" type="button" value="..."/>
<input checked="" type="checkbox"/>	<b>Accessibility checkpoint group</b> VPAT checkpoint group	Select List (sing...	2 projects	2 screens	June 17, 2020	1,654	<input style="background-color: #ccc; border: 1px solid #ccc; padding: 2px 5px;" type="button" value="..."/>
<input type="checkbox"/>	<b>Accessibility impact</b>	Text Field (singl...	0 projects	2 screens	Never	0	<input style="background-color: #ccc; border: 1px solid #ccc; padding: 2px 5px;" type="button" value="..."/>
<input type="checkbox"/>	<b>Actual Result</b>	Text Field (multi...	Global (all proje...	4 screens	February 12, 2021	6	<input style="background-color: #ccc; border: 1px solid #ccc; padding: 2px 5px;" type="button" value="..."/>

# Translating custom fields

You can translate the name and description of any custom field that you create into another language. You can only select from the language packs that are installed in Jira.

1. Go to **Administration > System**, and select **Custom fields**.
2. Find your custom field, and select **Actions** (  ) > **Translate**.
3. Select a language pack that this translation will belong to, and enter the translated strings for the name and description.

The translated strings will be displayed for the language packs youve chosen.

# Analyzing the usage of custom fields

**i** This functionality is available with the **Jira Data Center** license.

Having too many custom fields can clutter your Jira instance and affect its performance. Its normal for these fields to grow over time as admins create testing fields and duplicates, and some fields are just no longer used. However, monitoring and cleaning up your fields is necessary to keep your instance quick and healthy. This page explains some of the tools you can use to analyze the usage of your custom fields to decide if theyre still needed.

**i** For all of the following procedures, you must be logged in as a user with the **Jira System administrator** permissions. For details, see [Permissions overview](#).

## Finding custom fields that are no longer used

To view all of your custom fields, go to **Administration** (⚙️) > **Issues** and then select **Custom fields**. The **Issues** column shows you the number of issues that store a value for your field and includes both issues with default values and archived issues.

Name	Type	Available contexts	Screens	Last value update	Issues	Actions
<input type="checkbox"/> Accessibility impact	Text Field (single l...	0 projects	2 screens	Never	0	...
<input type="checkbox"/> Analytics required	Radio Buttons	0 projects	3 screens	Never	0	...
<input type="checkbox"/> Approver	User Picker (singl...	0 projects	9 screens	Never	0	...
<input type="checkbox"/> Approvers	User Picker (multi...	Global (all projects)	2 screens	January 19, 2021	0	...
<input type="checkbox"/> Beneficiary	Select List (single...	0 projects	1 screen	Never	0	...
<input type="checkbox"/> Borrowed on	Date Picker	0 projects	2 screens	Never	0	...
<input type="checkbox"/> Custom check boxes	Checkboxes	Global (all projects)	1 screen	Never	0	...

## Cleaning up

1. Click the **Issues** column name to sort in ascending or descending order.
2. Custom fields with a low number of issues are the most likely candidates for deletion. If you're not sure if a field should be deleted or not, use other columns described here to make your decision.

## Good to know

- Data in the **Issues** column is refreshed once a day. You can see the date of the last refresh above the column ("Issues: last refresh")
- Data won't be available for around 24 hours after the upgrade.

## Finding custom fields that haven't been updated for a long time

The **Last value update** column shows you the last time a value for a custom field has been added or updated in any issue, but it doesn't include default values as in, the user must actively choose or add a value for this field.

The screenshot shows the 'Custom fields' management page in Jira. A dropdown menu for 'Last value update' is open, displaying the following options: 'All', 'Never updated', 'More than 6 months ago', 'More than 1 year ago', 'More than 2 years ago', and 'Not updated since'. The background table lists custom fields with columns for Name, Type, Available contexts, Last value update, Issues, and Actions. The 'Last value update' column shows 'Never' for most fields, while 'Screens' are listed as '1 screen' or '2 screens'.

## Cleaning up

1. Filter your issues by **Last value update** to find custom fields that haven't been updated for a long time. We consider two years to be quite old, but this will depend on you.
2. Check the **Issues** column to see how many issues are using this field.
3. If the fields are outdated, but still used in thousands of issues, this might mean they're still important. If you're not ready to delete them, you could improve performance by narrowing down their context only to the projects they're used in.

## Good to know

- Data in this column is refreshed every hour, but the historical data about the usage of custom fields in existing issues won't be available for around 24 hours after the upgrade. Any new updates you make after upgrading will be available right away.
- If you're importing a project or just need to recalculate this data as a fallback method, you can recalculate it in Jira advanced settings. To open the settings, go to **Administration** (⚙️) > **System** and then select the **Advanced settings** button.

## Comparing to other columns to help you make the decision

**Issues** and **Last value update** are the best tools to identify fields that are a bit rusty, but not all such fields should be deleted. Some of them might store data from your past projects that you'd like to keep for reference. This is where other columns should help you.

### Available contexts

Shows specific projects that can use this field or shows **Global** if the field is allowed for all projects. This should help you decide how important your field is.

### Good to know

- Check the **Last value update** for your field. If it hasn't been used for a long time but you'd like to keep the data it contains, narrow it down to the projects it's been used in in the past.
- You can [change the contexts manually](#), or try to [optimize them automatically](#).

## Screens

Shows the screens where your custom field is displayed, like a Create issue screen or Default issue screen. The number of screens using your custom fields, and what screens those are, should help you decide whether it's really needed.

## More information

If you're on a mission to clean up your Jira instance and keep it that way, here are some useful pages:

- [Cleaning up your Jira instance](#)
- [Managing custom fields effectively](#)
- [Deleting custom fields](#)

# Optimizing custom fields

Custom fields can have a huge impact on the performance of your Jira instance. You can decrease this impact and speed up your Jira by improving the configuration of your custom fields. With the custom fields optimizer, you can do it automatically.

 Custom field optimization is available for Jira Software Data Center and Jira Service Management Data Center.

## Scanning your custom fields

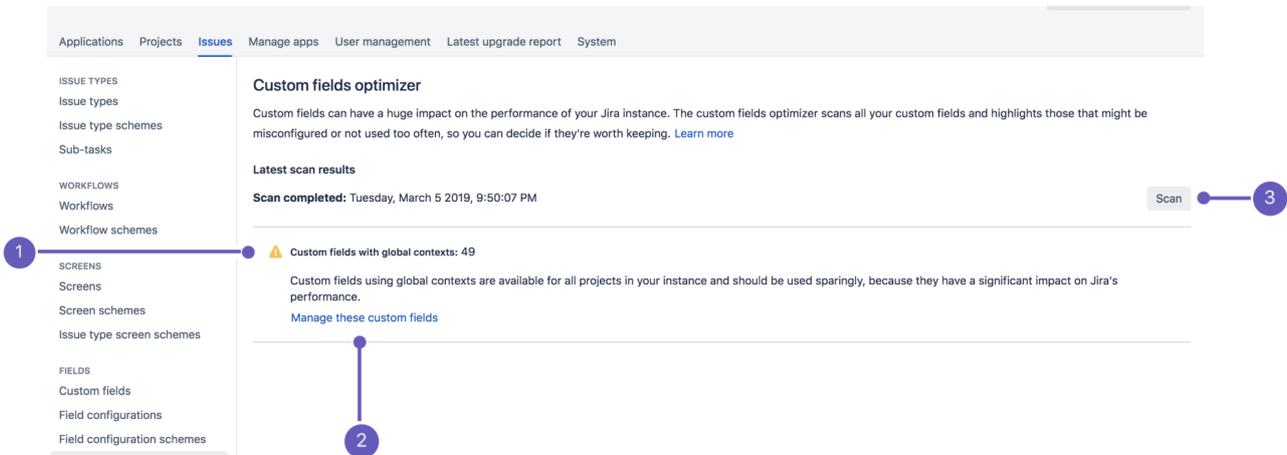
Scan your custom fields to find and highlight those whose configuration can be optimized. The scan will exclude any custom fields created automatically by Jira Software, Jira Service Management and Portfolio for Jira.

To scan your custom fields:

1. Log in as a Jira administrator, and go to **Administration** () > **Issues**.
2. In the menu on the left, click **Custom fields optimizer**.
3. Click **Scan**. Depending on the size of your instance, it might take some time. Just leave it running and come back later.

## Viewing the results

When the scan is complete, it will show you why custom fields were highlighted, and how many of them can be optimized.



1. **Recommendation:** Shows why custom fields are highlighted, and tells you how many fields can be optimized.
2. **Manage custom fields:** Takes you to the page with highlighted custom fields, where you can optimize them.
3. **Scan:** Action to start the scan, or to run it again.

## Custom fields with global contexts

These custom fields are available to all projects in your Jira instance, although only a handful actually use them. For more info on how to optimize these fields, see [Custom fields with global contexts](#).

## Custom fields and archiving issues

Archived issues are deleted from Jira index and because of that Custom Filed Optimizer does not have the full set of information about the issues which use a particular custom field. This can result in some unwanted behaviour such as custom fields not displaying for archived issues and for the issues that have been restored. For more information, see [Jira Knowledge Base](#).

# Custom fields with global contexts

Custom fields with global contexts are available to use by all projects in your Jira instance, but only a handful actually use them. To improve performance, you can automatically change their context to project-specific, applying them only to relevant projects.

**i** Custom field optimization is available for Jira Software Data Center and Jira Service Management Data Center.

## Viewing the custom fields

The optimizer creates a list of custom fields whose configuration is not optimal, and sorts them based on the performance impact they have on your Jira instance. Note, the list of custom fields will not include those created by Jira Software, Jira Service Management or Portfolio for Jira.

To be optimized, a custom field must have a global context and be used in several projects only. A good practice is for the field to be used in fewer than 10 projects. Otherwise, it's considered an actual global custom field, and you shouldn't change its configuration.

The list of custom fields looks like in the following example:

Field name	Type	Used by	Action
Status Reason	Text Field (single line)	10 projects	Change context
Support Hours Saved	Number Field	10 projects	Change context
Support Tickets Prevented	Number Field	2 projects	Change context
What does success/ failure look like?	Text Field (multi-line)	2 projects	Change context
Date Resolved	Checkboxes	10 projects	Change context
Date Created	Updated By Me	10 projects	Change context
Targets	Checkboxes	1 project	Change context
Affected Version(s)	Text Field (single line)	2 projects	Change context
Approved	User Picker (multiple users)	2 projects	Change context
JPOS - Bug Cause	Select List (single choice)	1 project	Change context
Capture for Jira jQuery Version	Capture for Jira text	2 projects	Change context

1. **Name** of a custom field.
2. **Used by**: View projects that are currently using this custom field.
3. **Actions**: Select **Change context** to change the configuration to an optimal one. Later, you can select **View change** to see the custom field's configuration page, and make any changes you need.

## Changing the context

The global context was the reason why your custom fields were highlighted, and the solution is to change the context to project-specific. After doing so, your custom fields will be applied only to projects that currently use and need them.

To change the context, click **Change context** next to each custom field.

## Best practices

Try applying the changes in batches and during off-peak hours. When one of your users opens an issue that uses a certain custom field, the cache will need to be recreated. If you do it in batches, the cache will be recreated once for multiple custom fields.

## Changes

After making the change, we'll take care of the following actions for you:

- We'll identify all projects that use a given custom field.
- We'll change the context from global to project-specific, and apply the custom field only to relevant projects (by adding these projects to the local context.)

 There's no risk of data loss, because we're not removing any values, or the custom field itself. Projects that use this custom field will continue to use it in the same way.

## Next steps

Click **View change** to see the custom field's context. You can check the list of projects that were added to the context, add some new ones, or change the context back to global if you don't like the outcome.

The only thing you have to keep in mind is to add any future projects to the list. They won't be able to use the custom field until they belong to its new context. For more info about editing contexts and managing associated projects, see [Configuring custom fields](#).

# Managing system fields

Every Jira instance provides a number of [out-of-the-box system fields](#). They are designed to collect and carry the information necessary to build issues. Most of the system fields are locked for editing. Jira admin can associate system fields with different screens and check available contexts.

**i** In Jira 8.16 we divided fields into system fields and custom fields. We've also enabled a possibility of adding contexts and default values to the **Description** field. This functionality is available with a **Data Center license**.

## Viewing system fields

To view and manage your custom fields:

1. Go to **Administration > Issues**.
2. Click **Custom fields**.

1	Name	Available contexts	Screens	Actions
	Description	2 4 Contexts	1,637 screens	3 ...
	Summary	-	1,810 screens	...
	Issue Type	-	1,769 screens	...
	Priority	-	1,598 screens	...
	Resolution	-	774 screens	...
	Affects Version/s	-	445 screens	...
	Fix Version/s	-	865 screens	...

4

1. **Columns:** All the columns to help you identify the field itself and the importance of it.
2. **Available contexts:** Displaying the contexts associated with the Description field. This view gives an easy way to edit available contexts, define new ones, and preset the default value for the Description field.
3. **Actions:** Configure your system fields by defining associated screens. For the Description field, you can also define contexts and the default values.
4. **Screens:** Displaying the screens associated with the field. This view gives you also an easy way to configure screens.

## Identifying your fields

Every field is described by a set of columns to help you identify your fields and their importance within your Jira instance.

Column name	Description
-------------	-------------

Name	Name of a system field.  See list of <a href="#">all available system fields</a>
Available contexts	Contexts are different configurations of the same field. For example, you can configure a field to use different values for different projects or issue types, or make it available only in some projects.  See <a href="#">Configuring contexts and default values for the Description field</a>
Screens	Screens are all these different forms that you fill in when creating issues, editing them, or transitioning them through workflows. Every field needs to be associated to some screens, or it wont be visible anywhere.  See <a href="#">Viewing and configuring screens</a>

 In Jira Service Management, default values for the Description field will be available when your agents create or edit requests, but they wont be displayed on customer portals.

## Managing system fields

What would you like to do with your custom fields?

- [Configuring contexts and default values for the Description field](#)
- [Viewing and configuring screens](#)

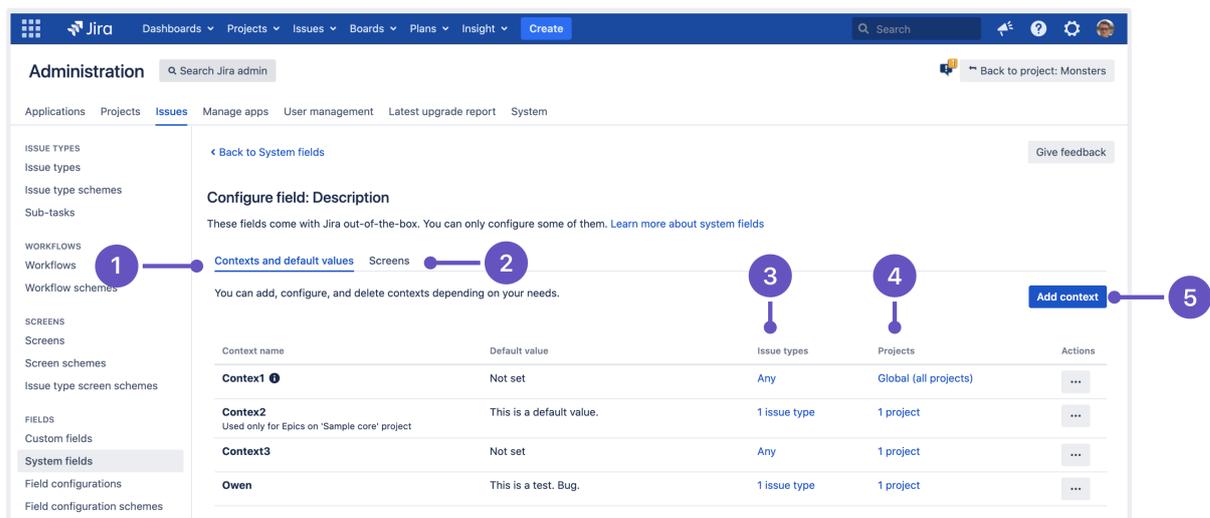
# Configuring contexts and default values for the Description field

## Viewing contexts for the Description field

 This functionality is available with a **Data Center license**.

To view existing contexts for your field and to add some new ones:

1. Go to **Administration > Issues**, and select **System fields**.
2. Find the Description field, and select **Actions > Contexts and default values**



1. **Context and default values** - Here you can view, edit, delete contexts, or go back to the list of system fields.
2. **Screens** - This view allows you to associate the Description field with screens and check the number of already associated screens.
3. **Issue types** - List of issues types where the Description field is being used.
4. **Projects** - List of projects where the Description field is being used.
5. **Add context** - Here you can add new contexts to the Description field.

## Adding a new context and default values for the Description field

To add a new context:

1. Select **System fields > Actions > Contexts and default values > Add context**.
2. Choose a name for this context and a helpful description.
3. Select the issue types and projects this context should apply to.
4. Specify the default value using available formatting and styles. You can use visual or text editor.
5. Click **Add**.

## Editing and deleting contexts and default values for the Description field

After you add a new context or a default value, you can change details, default values, and options:

- To change the details of this context, like name and description, and the issues and projects this context applies to, navigate to the context and select **Context and default values > Actions > Edit**.

If you want to delete a context, go to **Context and default values > Actions > Delete**

For detailed description and steps to reproduce to avoid data loss, see [Change context for an existing custom field without losing old data](#).

## Viewing and configuring screens

For Jira Data Center, you can apply screens to a system field. It's also possible to check the number of associated screens to a particular system field.

To view and configure all screens associated with a system field, go to **System Fields > Actions > Screens**.

# Specifying field behavior

A **field configuration** defines the behavior of *all fields* available in your Jira installation, including Jira's own 'fixed'/'built in' fields (known as 'system' fields) and [custom fields](#).

For each field, a field configuration specifies:

- the **description** that appears under the field when an issue is edited
- whether the field is **hidden** or **visible**
- whether the field is **required** (i.e. the field will be validated to ensure it has been given a value) or **optional**
- (for text fields only) which **renderer** to use

## On this page:

- [Managing multiple field configurations](#)
- [Modifying field behavior](#)

When defining field behavior for one or more Jira projects and the fields used by the issue types in these projects, you typically start by adding one or more new field configurations (see [below](#)). You then begin [modifying the behavior](#) of individual fields in these new field configurations.

 A new field configuration should be added for each project and issue type combination which requires specific fields to be present and/or fields that express unique behavior. You can then associate each new field configuration with a different issue type through a ['field configuration scheme'](#). A field configuration scheme can then be associated with one or more projects. For more information, please see the [Overview Diagram](#).

 For all of the following procedures, you must be logged in as a user with the [Jira Administrators global permission](#).

## Managing multiple field configurations

You can create multiple field configurations for use on separate projects and issue types.

- Multiple field configurations are organized into [field configuration schemes](#), which associate field configurations with issue types.
- A scheme can then be associated with one or more projects, allowing you to control fields on a per project, per issue type basis. See [Associating field behavior with issue types](#) for more information.

## About the 'Default Field Configuration'

When Jira is installed, the **Default Field Configuration** is created automatically. All new projects are associated with this configuration. This configuration is also used for projects that are not associated with a [field configuration scheme](#).

 It is not possible to delete the **Default Field Configuration**.

## Adding a field configuration

1. Choose **Administration**  > **Issues**.
2. Select **Fields > Field Configurations** to view all your field configurations.
3. Click the **Add New Field Configuration** button to open the **Add Field Configuration** dialog box.
4. Complete the **Add Field Configuration** dialog box:
  - **Name** enter a short phrase that best describes your new field configuration.
  - **Description** (*optional but recommended*) enter a sentence or two to describe when this field configuration should be used.
5. Click the **Add** button to add your new field configuration to Jira. Once you have added your new field configuration, you can then begin modifying the behavior of its fields ([below](#)).
  -  You will be taken directly to the **View Field Configuration** page, where you can modify the behavior of fields in your new field configuration. See [Modifying field behavior \(from step 4\)](#) below for details.

## Editing a field configuration

1. Choose **Administration**  > **Issues**.
2. Select **Fields > Field Configurations** to open the View Field Configurations page, which lists all your field configurations.
3. Click the **Edit** link next to the field configuration you wish to edit.
4. On the **Edit Field Configuration** page, edit the field configuration's **Name** and **Description**.

**Please note:** The **Default Field Configuration** cannot be edited.

## Deleting a field configuration

1. Choose **Administration**  > **Issues**.
2. Select **Fields > Field Configurations** to open the View Field Configurations page, which lists all your field configurations.
3. Click the **Delete** link next to the field configuration you wish to delete.
  - You will be prompted to confirm this operation.**

**Please note:**

- The **Default Field Configuration** cannot be deleted.
- You can only delete a field configuration that is not associated with a [field configuration scheme](#). The **Delete** link will not be available for field configurations which are associated with one or more field configuration schemes.

## Copying a field configuration

1. Choose **Administration**  > **Issues**.
2. Select **Fields > Field Configurations** to open the View Field Configurations page, which lists all your field configurations.
3. Click the **Copy** link next to the field configuration you wish to copy.
4. On the **Copy Field Configuration** page, specify the **Name** and **Description** for the field configuration to be copied.
  - The (initial) field settings between the original and copied field configurations will be identical.**

**Please note:** a newly created field configuration will not take effect until you:

1. [Associate your new field configuration to one or more issue types.](#)
2. [Associate that field configuration with one or more projects.](#)

See [Associating field behavior with issue types](#) for more information.

## Modifying field behavior

To modify the behavior of fields in Jira, you need to modify the field configurations that those fields have been defined in.

1. Choose **Administration**  > **Issues**.
2. Select **Fields > Field Configurations** to open the View Field Configurations page, which lists all your field configurations.
3. Locate the field configuration of interest and click the **Configure** link to open the **View Field Configuration** page, which lists all system and custom fields in your Jira installation for that field configuration.

**Please note:**

- The **Edit** link only allows you to change the **Name** and **Description** of the field configuration, not of individual fields.
  - Note that the **Edit** link is not available for the **Default Field Configuration** on the **View Field Configuration** page (listing all field configurations defined in your Jira installation).
4. In the **Operations** column, you can perform the following actions for any field:
    - **Edit** change the field's description (i.e. help text).
    - **Hide/Show** hide the field from view or show it.

- [Require/Optional](#) set a field to be required (so that whenever a field is edited it must be given a value) or optional.
- [Renderers](#) change a field's renderer (see [Configuring renderers](#) for more information).

**Please note:** a newly created field configuration will not take effect until you:

1. [Associate your new field configuration to one or more issue types](#), and then
2. [Associate that field configuration with one or more projects](#).

See [Associating field behavior with issue types](#) for more information.

### Editing a field's description

Fields can be given descriptions to better identify the meaning of the field. These descriptions are typically displayed under the fields they are associated with when creating or editing an issue.

1. Follow the first three steps above (in [Modifying field behavior](#)) to access the field configuration whose field's description you wish to edit.
2. Click the **Edit** link next to the field you want to change and update the field's description.
3. Click the **Update** button to save your changes.

### Hiding or showing a field

If your organization or project has no use for a particular field, you have the option to hide it. Hiding a field will ensure that the field does not appear on any screens (i.e. issue operation screens, workflow transition screens) where the field configuration applies.

**Please note:**

- Hiding a field in the field configuration is distinct from not adding a field to a screen. Fields hidden through the field configuration will be hidden in *all* applicable screens, regardless of whether or not they have been added to the screen.
- For fields that have a default value: If the field is hidden in the field configuration, then it will not receive a value when an issue is created, regardless of whether the field is present on the [Create Issue screen\(s\)](#). (The following fields can have a default value: [resolution](#), [status](#), [priority](#), [issue type](#), [security level](#), and [custom fields](#).)
- The fields **Summary** and **Issue Type** cannot be hidden and as such there is no **Hide** option available for these fields.
- If you hide the **Fix Version/s** field, the Change Logreport will not work.

1. Follow the first three steps above (in [Modifying field behavior](#)) to access the field configuration whose fields you wish to hide or show.
2. Do either of the following:
  - If you no longer want to expose a field through Jira's user interface, click the **Hide** link associated with that field.
    - You can make this field visible again at any time by clicking the **Show** link.
  - If you want to show a field (which is currently hidden) through Jira's user interface, click the **Show** link associated with that field.
    - You can hide this field again at any time by clicking the **Hide** link.

### Making a field required or optional

Certain fields within your organization may be compulsory for issues. In this case you can set a field to be required, so that Jira validates that the field has been given a value whenever an issue is edited. If a required field has not been given a value, Jira will return an error informing the user that the field should be filled.

1. Follow the first three steps above (in [Modifying field behavior](#)) to access the field configuration whose fields you wish to hide or show.
  - When viewing a field configuration (see [above](#)), fields which are already required have that indication next to their name.
2. Do either of the following:

- To make a field mandatory when used through Jira's user interface, click the **Required** link associated with that field.
  - ⓘ The text **Required** will appear next to the field's name.
- To make a field optional, click the **Optional** link associated with that field.
  - ⓘ The **Required** text next to the field's name will disappear.

ⓘ **Please note:**

- Fields that are hidden cannot be set to required.
- If you make a field **Required**, ensure that the field is present on your **Create Issue** screen(s).
  - Note that you can have different field configurations for different projects and issue types (see [Associating field behavior with issue types](#)), so you need to ensure that all **Required** fields are present on the **Create Issue** screens for all associated projects and issue types (see [Associating screen and issue operation mappings with an issue type](#)).
  - Be aware that there is a feature request ([JRA-5783](#)) to make a field required for only one transition. If you are interested, please watch that issue for status updates.

### Changing a field's renderer

Before you change the renderer for a specific field, please read [Configuring renderers](#), paying particular attention to the [Implications for Jira operations](#) section.

1. Follow the first three steps above (in [Modifying field behavior](#)) to access the field configuration whose field's renderer you wish to change.
  - ⓘ When viewing a field configuration (see [above](#)), the **Name** column indicates which renderers are currently enabled for all renderable fields, with the current renderer shown in brackets immediately below its field name.
2. Click the **Renderers** link for the field you want to change. This will take you to a page where you will have the option to select a renderer from all configured and available renderers.
3. This page will warn you if there are issues that will be affected by the change. If no issues will be affected then the warning does not show. From this page, choose the renderer you wish to use and click **Update**.

ⓘ Changing the renderer only affects how a Jira field's content is *displayed* or how a user *interacts* with a multi-select field it does not affect the issue data that exists in the system. Hence, you can therefore toggle between renderer types safely.

# Associating field behavior with issue types

A **field configuration scheme** associates (or "maps") [field configurations](#) to issue types in a project. In turn, a field configuration scheme can be [associated](#) with one or more projects.

This means that you can define different [field configurations](#) for each issue type that is available in a given project. For example, it is possible to have separate field configurations for the **Bug** the **Improvement** issue types (whose associations are defined in a field configuration scheme) for a project called 'Test'. Refer to the [Overview Diagram](#) for more information.

Because a field configuration scheme can be associated with more than one project (and associations between field configurations and issue types in a field configuration scheme are flexible), you can minimize your administrative workload as you can reuse the same field configuration for the same (or different) issue types across multiple projects.

**Note:**For all of the following procedures, you must be logged in as a user with the [Jira Administrators](#) global permission.

- [Adding a field configuration scheme](#)
- [Editing a field configuration scheme](#)
- [Deleting a field configuration scheme](#)
- [Copying a field configuration scheme](#)
- [Associating a field configuration scheme with a project](#)

## Adding a field configuration scheme

1. Choose **Administration**  > **Issues**.
2. Select **Fields** > **Field Configurations** to view all your field configurations.
3. Click the **Add New Field Configuration Scheme** button to open the **Add New Field Configuration Scheme** dialog box.
4. Complete the **Add New Field Configuration Scheme** dialog box:
  - **Name** enter a short phrase that best describes your new field configuration scheme.
  - **Description** (*optional but recommended*) enter a sentence or two to describe when this field configuration scheme should be used.
5. Click the **Add** button to add your new field configuration to Jira.
  - You will be taken directly to the **Configure Field Configuration Scheme** page, where you can start associating issue types with field configurations in your new field configuration scheme. See [Modifying field behavior \(from step 4\)](#) for details.

## Associating an issue type with a field configuration

1. Choose **Administration**  > **Issues**.
2. Select **Fields** > **Field Configurations** to open the View Field Configurations page, which lists all your field configurations.
3. Click the **Configure** link for the [field configuration scheme](#) in which to create an association between an a field configuration and an issue type. The **Configure Field Configuration Scheme** page will appear, showing the scheme's current mappings of field configurations to issue types.
  - If you have not added any new field configurations since installing Jira, you will only have Jira's **Default Field Configuration** to work with.
4. Click **Associate an Issue Type with a Field Configuration**.
5. Select the desired issue type and field configuration and click the **Add** button.

### Please note:

- An issue type can only have one association within a given configuration scheme.
- If an issue type does not have an association in the scheme, the field configuration associated with the **Default** entry in the scheme will be used for issues of that type.

## Removing an association between an issue type and a field configuration

1. Choose **Administration**  > **Issues**.
2. Select **Fields** > **Field Configurations** to open the View Field Configurations page, which lists all your field configurations.

3. Click the **Configure** link for the [field configuration scheme](#) that contains the association between a field configuration and issue type you want to remove. The **Configure Field Configuration Scheme** page will appear, showing the scheme's current mappings of field configurations to issue types.
  - 📘 If you have not added any field configurations since installing Jira, you will only have Jira's **Default Field Configuration** to work with.
4. Click the **Remove** link next to the issue type you wish to remove from the scheme.

📘 **Please note:** The **Default** entry cannot be removed from the scheme.

### Associating an issue type with a different field configuration

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Fields > Field Configurations** to open the View Field Configurations page, which lists all your field configurations.
3. Click the **Configure** link for the field configuration scheme contains an association between a field configuration and issue type you want to change. The **Configure Field Configuration Scheme** page will appear, showing the scheme's current mappings of field configurations to issue types.
  - 📘 If you have not added any field configurations since installing Jira, you will only have Jira's **Default Field Configuration** to work with.
4. Click the **Edit** link next to the issue type whose field configuration you wish to change.
5. Select the new **Field Configuration** you would like to associate with this issue type.
6. Click the **Update** button.

### Editing a field configuration scheme

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Fields > Field Configurations** to open the View Field Configurations page, which lists all your field configurations.
3. Click the **Edit** link next to the [field configuration scheme](#) whose name and description you wish to modify.
4. On the **Edit Field Configuration Scheme** page, edit the **Name** and **Description** of the field configuration scheme.
5. Click the **Update** button.

### Deleting a field configuration scheme

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Fields > Field Configurations** to open the View Field Configurations page, which lists all your field configurations.
3. Click the **Delete** link next to the [field configuration scheme](#) you wish to delete. You will be prompted to confirm your deletion.

📘 You can only delete a field configuration scheme that is not associated with a [project](#). The **Delete** link will not be available for field configuration schemes which are associated with one or more projects.

### Copying a field configuration scheme

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Fields > Field Configurations** to open the View Field Configurations page, which lists all your field configurations.
3. Select **Administration > Issues > Fields > Field Configuration Schemes** (tab) to open the **View Field Configuration Schemes** (above), which lists all your field configuration schemes (if any exist).
4. Click the **Copy** link next to the [field configuration scheme](#) you wish to copy.
5. On the subsequent page, specify the **Name** and **Description** of the field configuration scheme to be copied.
6. Click the **Copy** button.
  - 📘 The (initial) associations between field configurations and issue types in both the original and copied field configuration schemes will be identical.

### Associating a field configuration scheme with a project

To make your Jira projects use your field configurations, you need to associate these field configuration(s) with issue types in a field configuration scheme ([above](#)) and then associate this field configuration scheme with a project. (This association means that the field configuration scheme will be applied to the project.) Once this is done:

- The issues in this project will use the field configuration(s) 'mapped' to their issue type (defined by the field configuration scheme associated with the project)  
*but also:*
- The issue types available to this project are defined by the [issue type scheme](#) associated with the project.

Therefore, even though a project's field configuration scheme may associate various different field configurations with a large set of issue types, only a subset of these issue types (as defined by the project's issue type scheme) and hence, field configurations themselves, may be available in that project. In other words, the issue types available to a project are restricted by the project's issue type scheme.

**i** Note that newly created projects are not associated with any field configuration schemes and hence, use the [Default Field Configuration](#) for all issues.

#### To associate a field configuration scheme with a project:

1. Access the **Project Summary** administration page for your project (see [Configuring a project](#)).
2. In the **Fields** section of this page, click the name of the current [field configuration scheme](#).
3. Click the **Actions** dropdown menu and choose **Use a different scheme**.
4. In the resulting page, select the scheme you want to associate with this project.  
**i** Selecting *None* will result in all issue types available to your project using Jira's [Default Field Configuration](#).
5. Click the **Associate** button. You will be returned to the [Project Summary administration page](#), with the project now associated with the selected field configuration scheme.

# Configuring renderers

Renderers are configured on a per field basis. To configure a renderer for a particular field, see [Specifying field behavior](#). Note that you can configure the same field differently for different projects and issue types see [Associating field behavior with issue types](#).

Renderers are implemented as Jira apps, meaning that any renderer can be easily added to or removed from use within Jira. This includes any custom renderers that may be developed.

Please read [Implications for Jira operations](#) below before configuring renderers.

## On this page:

- [Renderable fields](#)
- [Renderer types](#)
- [Implications for Jira operations](#)
- [Configuring renderers](#)

**i** Renderers affect the rendering (view) of a field's value. This means that you can migrate to a different renderer without affecting your issue data; only the view will be changed. It also means that if you do not like the way your issues look using the new renderer, you can simply switch back with no impact on your issue data.

For all of the following procedures, you must be logged in as a user with the **Jira Administrators** [global permission](#).

## Renderable fields

Potentially any field within Jira applications can be a renderable field, but this only really makes sense in the case of text-based fields (for the default text renderer and the wiki style renderer) and multi-select fields (for the autocomplete renderer and the select list renderer). The following table shows the Jira fields that are renderable out-of-the-box:

Field	Available Renderers
Description	<a href="#">Wiki style renderer</a> (default), <a href="#">Default text renderer</a>
Comment	<a href="#">Wiki style renderer</a> (default), <a href="#">Default text renderer</a>
Environment	<a href="#">Wiki style renderer</a> (default), <a href="#">Default text renderer</a>
Component	<a href="#">Autocomplete renderer</a> (default), <a href="#">Select list renderer</a>
Affects version	<a href="#">Autocomplete renderer</a> (default), <a href="#">Select list renderer</a>
Fix version	<a href="#">Autocomplete renderer</a> (default), <a href="#">Select list renderer</a>
Custom field of type "Free Text Field (unlimited text)"	<a href="#">Wiki style renderer</a> (default), <a href="#">Default text renderer</a>
Custom field of type "Text Field"	<a href="#">Wiki style renderer</a> (default), <a href="#">Default text renderer</a>
Custom field of type "Multi Select"	<a href="#">Select list renderer</a>
Custom field of type "Version Picker"	<a href="#">Autocomplete renderer</a> (default), <a href="#">Select list renderer</a>

## Renderer types

Jira ships with the following renderers:

- for text fields: [wiki style renderer](#) and [default text renderer](#)
- for multi-select fields: [autocomplete renderer](#) and [select list renderer](#)

## Default text renderer

The default text renderer renders a field's content as plain text, with the following additional auto-linking features:

Content	Description	Sample
Jira issues	If the text contains text that resolves to a Jira issue key, an HTML link will be generated, pointing to that issue.	<div style="border: 1px solid black; padding: 5px;"> <p><b>Description</b> _____</p> <p>This relates to <a href="#">ANGRY-304</a></p> </div>
Web links	If the text contains text that resolves to a web page on a website, an HTML link will be generated, pointing to that web link.	<div style="border: 1px solid black; padding: 5px;"> <p><b>Description</b> _____</p> <p>More details are found in <a href="https://answers.atlassian.com">https://answers.atlassian.com</a>.</p> </div>

 It is not possible to disable the default text renderer app as it is required for the system to function properly. If a text field is setup to use a renderer that is later disabled, the field will revert to using the default text renderer.

### Wiki style renderer

The wiki style renderer allows a user to enter wiki markup to produce HTML content.

This renderer uses the Confluence wiki renderer engine and therefore uses the Confluence wiki notation. The Confluence notation is easy to learn and allows for:

- Italic, bold and underlined text
  - Multiple levels of headings to organize your document
  - Bullets, numbering, tables and quotations
  - Images, screenshots, and emoticons
  - Powerful mini-applications using macros
- A full notation guide can be found [here](#).

 The wikistyle renderer can only be used with JDK 1.4 and up. The renderer will not run on JDK 1.3.

Please note that some fields may require further field behavior configurations to be enabled see [Specifying field behavior](#).

### Wiki style renderer macro support

The Wiki style renderer supports pluggable macros in the same way that Confluence does. Macros provide an easy and powerful extension point to the wiki markup language. Jira ships with a number of macros.

Jira and Confluence can share macros, but keep in mind that many Confluence macros are very specific to the Confluence application and will therefore not run within Jira. For example, the 'children' macro in Confluence shows links to all of a page's child pages. Jira has no concept of 'page', and therefore, this macro will not function in Jira.

### Autocomplete and select list renderers

The autocomplete and select list renderers let you start typing text, which is then autocompleted, or to select

The screenshot shows a form field with the label "Fix Version/s:". The input field contains the text "1.". A dropdown menu is open below the input field, displaying a list of "Released Versions". The list includes "1.3", "1.2", and "1.1". The option "1.3" is currently selected and highlighted in blue.

from a drop-down list of options:

## Implications for Jira operations

The fact that Jira allows you to configure different renderers across different projects/issue types for the same field has implications for bulk operations. Also, since the wiki style renderer inherently creates HTML as its end product, there are implications as to how this will behave when issue data is viewed outside Jira's web front-end.

### Bulk move

When performing a bulk move operation you can either move issues to an environment (project/issue type) where the renderer types for the fields are the same or where they will be different.

#### If the renderer types are the same

If the renderer types for where you are moving to are the same then you will not notice any changes to the way the issues data is displayed once the move has occurred and the move operation will not prompt you with any warnings.

#### If the renderer types are different

When bulk moving issues to an environment (project/issue type) that has a different renderer type defined for one of the fields being affected by the move, if any of the issues have a non empty value associated with the field, the move operation will present you with a warning so that you are aware of the change. The warning does not affect the move operation in any way but it is there to alert you to the fact that the moved issues' affected fields may look different in their new project/issue type.

### Bulk Edit

When performing a bulk edit operation the only renderable fields you may be able to bulk edit are instances of the text field, and free text field(unlimited text) custom fields. The bulk edit operation does not allow you to bulk edit the description, environment, or comment fields.

**You will only be allowed to bulk edit a renderable field if all the issues selected for edit use the same renderer type.** If the renderer type differs for any of the selected issues you will be presented with an error message.

This is best illustrated with an example. Let's say you have two global custom fields, 'Custom text area' and 'Custom text field', whose types are as their names imply. Let's say you have project 'A' which is configured to use the wiki style renderer for both of the fields. Let's say you also have a project 'B' which is configured to use the default text renderer for the 'Custom text area' field and the wiki style renderer for the 'Custom text field'. Let's also say that you have one issue in each project. If you were to perform a bulk edit operation on the two issues in these projects, you will be presented with the following:



Jira allows the Issue Navigator view to be exported to RSS/XML. If a field is using the default text renderer its values will be exported in a CDATA section within the generated XML. If a field is using the wiki style renderer, its rendered value will be XML escaped and included in the generated XML. If the XML view is being used as an RSS feed, most RSS readers will render the generated HTML so you will see the rich content within your RSS reader.

If you would like to have this view feed out the raw values (unrendered) then you can send an additional request parameter 'rssMode=raw'. If the original link looks like this:

```
http://localhost:8080/browse/AAA-1?decorator=none&view=rss
```

Then the URL to have the raw values placed inside a CDATA should look like this:

```
http://localhost:8080/browse/AAA-1?decorator=none&view=rss&rssMode=raw
```

## Editing a renderable custom field's default value

When editing a renderable custom field's default value, even if it is only ever configured to use the wikistyle rendereryou will not be presented with the edit and preview options. Unfortunately, in this context, it is not possible to tell which renderer should be used for editing. However, if you enter a default value using wiki markup, then this will render correctly in environments (project/issue type) where the field has been configured to use the wikistyle renderer.

## Configuring renderers

### Applying a renderer to a field

To enable a renderer for a particular field, edit the field configuration, and choose the appropriate renderer for the field. For details, see [Specifying field behavior](#).

### Enabling a renderer app

Renderers within Jira are implemented as Jira apps. The macros that the wikistyle rendereruses are also implemented as Jira apps. For general information on apps, see the [Jira app Guide](#).

Note that apps are configured at an instance-wide level it is not possible to configure apps at a project/issue type level.

### Configuring a renderer app

Renderers and their dependant components, except for the default text renderer, can be enabled or disabled as follows.

1. Choose **Administration**  > **Manage apps**.
2. The 'Find apps' screen shows apps available via the [Atlassian Marketplace](#). Choose **Manage apps** to view the apps currently installed on your Jira instance.
3. Select **Manage apps**, and then search for 'renderer', filtering for system apps, as shown here:

This screen displays all the configured renderers within Jira.

- Click the **Disable** button to deactivate the renderer for the entire instance of Jira.

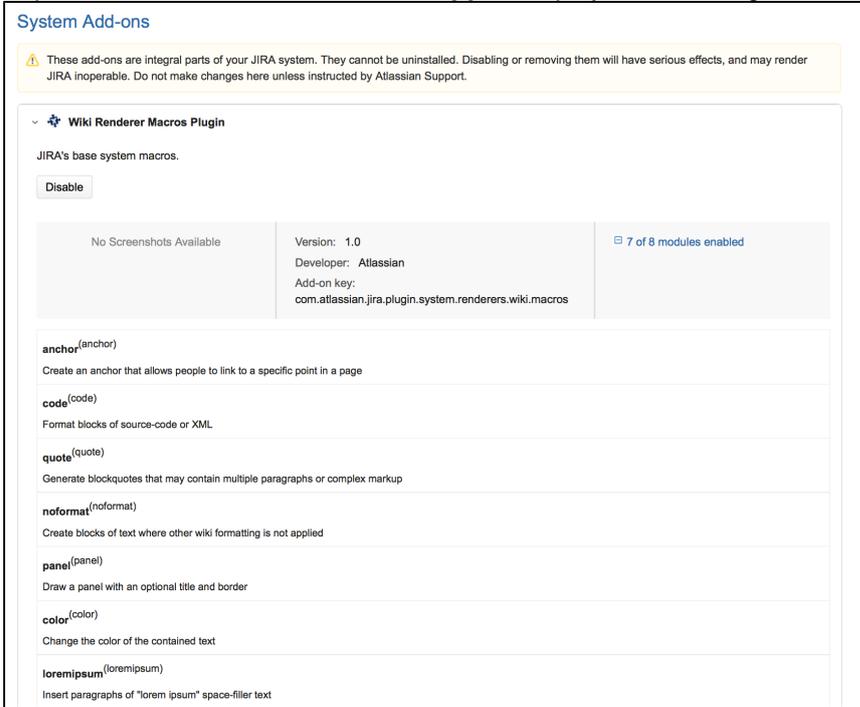
Any fields still set up to use a disabled renderer will fall back to the default text renderer. When you attempt to edit the field, a warning message alerts you to the fact that you are configured to use a renderer that is not available.

When a renderer is disabled it will not be available for selection when changing a field's renderer. To enable the renderer, click the **Enable** button. Enabling or disabling a renderer has no effect on the renderer settings in the field configurations, so it is possible to disable and then re-enable a renderer without affecting any data.

## Configuring macro apps for the wiki style renderer

The macros used by the wikistyle renderer can be enabled or disabled as follows.

1. Choose **Administration**  > **Manage apps**.
2. The 'Find apps' screen shows apps available via the [Atlassian Marketplace](#). Choose **Manage apps** to view the apps currently installed on your Jira instance.
3. Select **Manage apps**, and then search for 'renderer', filtering for system apps.
4. Expand the **Wiki Renderer Macros app** to display the following:



The screenshot shows the 'System Add-ons' configuration page for the 'Wiki Renderer Macros Plugin'. At the top, there is a warning message: 'These add-ons are integral parts of your JIRA system. They cannot be uninstalled. Disabling or removing them will have serious effects, and may render JIRA inoperable. Do not make changes here unless instructed by Atlassian Support.' Below this, the plugin is listed with a 'Disable' button. A summary bar shows 'No Screenshots Available', 'Version: 1.0', 'Developer: Atlassian', 'Add-on key: com.atlassian.jira.plugin.system.renderers.wiki.macros', and '7 of 8 modules enabled'. A list of macros is shown below, each with its name and a brief description:

- anchor**(anchor): Create an anchor that allows people to link to a specific point in a page
- code**(code): Format blocks of source-code or XML
- quote**(quote): Generate blockquotes that may contain multiple paragraphs or complex markup
- noformat**(noformat): Create blocks of text where other wiki formatting is not applied
- panel**(panel): Draw a panel with an optional title and border
- color**(color): Change the color of the contained text
- loremipsum**(loremipsum): Insert paragraphs of "lorem ipsum" space-filler text

From this screen you will see all the configured macros within Jira. If a macro is disabled then it will not be available to the wiki renderer. If you deploy any additional macros that you wish to use, they must be enabled here to be available to the wiki renderer. For more information on writing apps, see the documentation on [Writing Macros](#).

# Defining a screen

**Screens** group all available fields (or a subset of all available fields) defined in Jira applications, and organize them for presentation to a user. Through screens, you can control what fields are displayed to the user during [issue operations](#) (e.g. **Create Issue** and **Edit Issue** dialog boxes) or [workflow transitions](#) (e.g. **Resolve Issue** dialog box), as well as define the order in which these fields are shown to them. A screen also allows you to split subsets of fields across multiple tabs.

## On this page:

- [Adding a screen](#)
- [Editing a screen's details](#)
- [Copying a screen](#)
- [Deleting a screen](#)
- [Configuring a screen's tabs and fields](#)
- [Activating a screen](#)

When it comes to field visibility, screens functionally overlap slightly with [field configurations](#). For example, on the **Create Issue** dialog box, users will only see issue fields that:

1. are present on the screen associated with the issue's **Create Issue** issue operation
2. are *also not hidden* in the field configuration applicable to the issue (as defined by the project's [field configuration scheme](#))
3. the user has [permission](#) to edit (e.g. the **Due Date** field can only be edited by users with the **Schedule Issues** [project permission](#))

Hence, a field may be present on a screen used by a project, but if that field is hidden in the field configuration used by the project, that field will not be visible to the user when that screen in the project is displayed.

✔ If a particular field needs to be hidden at all times, it is easier to hide the field in the relevant field configuration than remove it from all screens.

⚠ Be aware that any newly created screen in Jira is not usable by a Jira project until it has been associated with either:

- An issue operation and issue type (via a [screen scheme](#) and then [issue type screen scheme](#))  
OR
- A workflow transition.

See [Activating a screen](#) (below) for details.

Jira applications ship with the **Default Screen**, **Resolve Issue Screen** and **Workflow Screen**, which are used as described below:

- **Default Screen** used for the default issue operations for creating, editing or viewing an issue.
- **Resolve Issue Screen** used for the transition view for the default **Close Issue** and **Resolve Issue** transitions, originating from the **Open**, **In Progress** and **Reopened** steps in Jira's default workflow.
- **Workflow Screen** used for the transition view for the default **Reopen Issue** transitions, originating from the **Resolved** and **Closed** steps and **Close Issue** transition, originating from the **Resolved** step in Jira's default workflow. The **Workflow Screen** defines a smaller set of fields than the **Resolve Issue Screen**.

## Adding a screen

### To add a new screen to Jira:

1. Log in as a user with the **Jira Administrators** [global permission](#).
2. Choose **Administration** (⚙️) > **Issues**. Select **Screens** to open the View Screens page, which lists all screens that have been defined in JIRA.
3. Click the **Add New Screen** button to open the **Add New Screen** dialog box.
4. Complete the **Add New Screen** dialog box:
  - **Name** enter a short phrase that best describes your new screen.
  - **Description** enter a sentence or two to describe the situations screen will be used.

- Click the **Add** button to add your new screen to Jira.
  - You will be taken directly to the **Configure Screen** page, where you can add fields to your new screen. See the [Configuring a screen's fields](#) section below for details.

## Editing a screen's details

### To change a screen's name and/or description:

- Log in as a user with the [Jira Administrators global permission](#).
- Choose **Administration** (⚙️) > **Issues**. Select **Screens** to open the View Screens page, which lists all screens that have been defined in JIRA.
- Click the **Edit** link next to the appropriate screen.
- You will now be directed to the **Edit Screen** page where you can edit the name and/or description of the Screen.

## Copying a screen

- Log in as a user with the [Jira Administrators global permission](#).
- Choose **Administration** (⚙️) > **Issues**. Select **Screens** to open the View Screens page, which lists all screens that have been defined in JIRA.
- Click the **Copy** link next to the Screen you wish to copy. You will be directed to the **Copy Screen** page, where you can enter a name and a description for the new Screen.

## Deleting a screen

- Log in as a user with the [Jira Administrators global permission](#).
- Choose **Administration** (⚙️) > **Issues**. Select **Screens** to open the View Screens page, which lists all screens that have been defined in JIRA.
- Click the **Delete** link next to the screen you wish to delete. You will be prompted to confirm your deletion.

**ⓘ** Screens that are associated with one or more screen schemes, or one or more workflow transitions, cannot be deleted.

## Configuring a screen's tabs and fields

You can configure the fields that display on a particular screen by adding/removing fields, as well as reordering them. Tabs can also be used to help group related fields. Tabs are useful for organizing complex screens, as you can place less used fields onto separate tabs. You can also add, remove and reorder tabs, as well as rename them.

### To configure a screen's tabs and fields:

- Log in as a user with the [Jira Administrators global permission](#).
- Choose **Administration** (⚙️) > **Issues**. Select **Screens** to open the View Screens page, which lists all screens that have been defined in JIRA.
- Click the **Configure** link next to the screen you want to add a field to. You can perform the following operations:

Operation	Instructions
Add a tab	Click <b>Add Tab</b> . Enter the name of the new tab in the dialog that appears and click <b>Add</b> .
Move a tab	Hover over the dotted part of the tab (next to the tab name) and drag the tab to the desired position.

Rename a tab	1. Hover over the tab name and click the <b>pencil icon</b> . 2. Enter the new name and click <b>OK</b> .
Delete a tab	Hover over the tab name and click the <b>X</b> .
Add a field	1. Click the tab that you want to add the field to. 2. Type the name of the field in the drop-down displayed at the bottom of the current fields. Field suggestions will appear as you type. 3. Click <b>Add Field</b> to add it to the current tab.
Move a field	Hover over the dotted part of the field (next to the field name) and drag the field to the desired position. Move a field to a different tab by dragging it to the name of the tab and dropping it.
Delete a field	Hover over the field and click the <b>Delete</b> button that appears.

### Tips on configuring screens

- **Date fields on View Issue screen** Fields of type 'Date' will always be displayed in the 'Dates' area of the default 'View Issue' screen, regardless of how you reorder them. This applies even if the dates are custom fields.
- **System fields on View Issue screen** System fields on the default 'View Issue' screen (e.g. Summary, Security Level, Issue Type, etc.) are fixed. This means that they will always appear in the same place on the 'View Issue' screen, even if you configure the Screen to move them onto a separate tab. Custom fields of related to Dates and People will also appear in their fixed section of the view issue screen. If none of the fields on a tab contain data then the tab is not shown. To make a tab show up, make sure it has a custom field with a type such as Text or Select and that the field has a value. Note, this information only applies to the screen associated with the 'View Issue' operation in a screen scheme.
- **Timetracking** You can add the ability to log work and/or specify/modify time estimates to a screen by adding the special **Log Work** and/or **Time Tracking** fields respectively.
  - If these fields cannot be found in the **Add Field** selection box and they have not already been added to the screen, check whether Jira's [Time Tracking feature](#) has been enabled. These fields will not be available to add to any screen if Time Tracking is disabled.
  - If any screens have the **Log Work** or **Time Tracking** fields and Jira's Time Tracking feature is subsequently deactivated, those screens will retain these fields until you specifically remove them. However, the fields will not be visible to the user until Time Tracking is reactivated.
- **Renaming standard Jira fields** You cannot rename the standard Jira fields (e.g. Priority, Summary, etc) via the Jira administration console. If you want to rename the standard Jira fields, you will need to modify files in your Jira installation. Please see [this knowledge base article](#) for instructions. Note, renaming the standard Jira fields is not supported.

### Activating a screen

To make a Screen available to users, you can **either**:

- Associate the Screen with an **issue operation** (e.g. 'Create Issue'), via a **Screen Scheme** see [Associating Screens with Issue Operations](#); or
- Associate the Screen with a **Workflow Transition** (e.g. 'Resolve Issue') see [Configuring Workflow](#).

# Associating a screen with an issue operation

## What is a 'screen scheme'?

A 'screen scheme' allows you to choose which [screen](#) will be shown to a Jira user when they perform a particular *issue operation*. There are three issue operations for which you can choose a screen:

- **Create issue** the screen that is shown when an issue is being created.
- **Edit issue** the screen that is shown when an issue is edited.
- **View issue** the screen that is shown when a user views an issue.

In a screen scheme, you can specify the same screen (or choose different screens) for these issue operations. Once you have created your screen scheme, you will need to activate it by associating the screen scheme with issue types via an '[issue type screen scheme](#)'. (In turn, issue type screen schemes are associated with Jira projects.)

 Please be aware that although it is possible to associate any screen defined in your Jira installation with either a screen scheme or a [workflow transition view](#), screen schemes and workflow transition views are distinct and unrelated.

**Note:** For all of the following procedures, you must be logged in as a user with the [Jira Administrators global permission](#).

## Adding a screen scheme

Depending on your requirements, you may want to create multiple screen schemes, and associate them with different projects and issue types.

1. Choose **Administration** () > **Issues**.
2. Select **Screens > Screen Schemes** to open the View Screen Schemes page.
3. Click the **Add New Screen Scheme** button.
4. Fill out the details for the new screen scheme on the form that is displayed.

**Note:** The default screen is used for issue operations that do not have a screen associated with them.

## Editing a screen scheme's details

1. Choose **Administration** () > **Issues**.
2. Select **Screens > Screen Schemes** to open the View Screen Schemes page.
3. Click **Edit** next to the desired screen scheme.
4. You will now be directed to the **Edit Screen Scheme** page, where you can edit the screen scheme's name and description and the Screen that is associated with the *Default Entry* of the scheme.

## Deleting a screen scheme

Note that screen schemes that are associated with an issue type screen scheme cannot be deleted. You will first need to edit the issue type screen scheme and remove the screen scheme.

1. Choose **Administration** () > **Issues**.
2. Select **Screens > Screen Schemes** to open the View Screen Schemes page.
3. Click the **Delete** link next to the desired screen scheme. You will be prompted to confirm your deletion.

## Copying a screen scheme

1. Choose **Administration** () > **Issues**.
2. Select **Screens > Screen Schemes** to open the View Screen Schemes page.

### On this page:

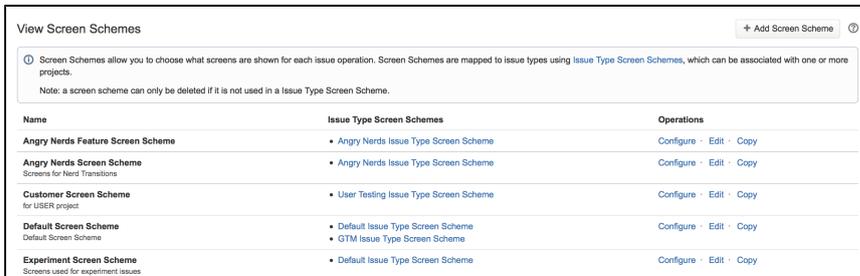
- [What is a 'screen scheme'?](#)
- [Adding a screen scheme](#)
- [Editing a screen scheme's details](#)
- [Deleting a screen scheme](#)
- [Copying a screen scheme](#)
- [Configuring a screen scheme](#)
- [Activating a screen scheme](#)

3. Click **Copy** next to the screen scheme you wish to copy.
4. You will now be directed to the Copy Screen Scheme page. Enter the name and description of the new screen scheme and click the **Copy** button.

## Configuring a screen scheme

### Associating a screen with an issue operation

1. Choose **Administration**  > **Issues**.
2. Select **Screens** > **Screen Schemes** to open the View Screen Schemes page:



Name	Issue Type Screen Schemes	Operations
Angry Nerds Feature Screen Scheme	• Angry Nerds Issue Type Screen Scheme	Configure · Edit · Copy
Angry Nerds Screen Scheme Screens for Nerd Transitions	• Angry Nerds Issue Type Screen Scheme	Configure · Edit · Copy
Customer Screen Scheme for USER project	• User Testing Issue Type Screen Scheme	Configure · Edit · Copy
Default Screen Scheme Default Screen Scheme	• Default Issue Type Screen Scheme • GTM Issue Type Screen Scheme	Configure · Edit · Copy
Experiment Screen Scheme Screens used for experiment issues	• Default Issue Type Screen Scheme	Configure · Edit · Copy

3. Locate the screen scheme in which you are interested, and click the **Configure** link next to it.
4. Click **Associate an Issue Operation with a Screen**, and select the following options:
  - a. Select the Issue Operation with which you wish to associate a screen.
  - b. Select the desired screen.

### Important notes

1. There can only be one association for an issue operation per screen scheme. If all operations have been associated with a screen, use the **Edit** link next to each operation to change the screen it is associated with.
2. If an issue operation does not have a specific mapping to a screen, the screen that is associated with the *Default* entry will be used for that operation. The *Default* entry cannot be deleted from a screen scheme. Click **Edit** next to the *Default* entry to change the screen that is associated with it.
3. The View Issue operation only allows you to control the layout of **custom fields** in the middle of the View Issue page. It ignores all the non-custom fields on the screen.

### Editing an association

1. Choose **Administration**  > **Issues**.
2. Select **Screens** > **Screen Schemes** to open the View Screen Schemes page.
3. Locate the screen scheme in which you are interested, and click the **Configure** link next to it. The Configure Screen Scheme page is displayed.
4. Click **Edit** next to the issue operation you wish to edit. The Edit Screen Scheme Item page is displayed.
5. Select the desired screen and click **Update**.

### Deleting an association

1. Choose **Administration**  > **Issues**.
2. Select **Screens** > **Screen Schemes** to open the View Screen Schemes page.
3. Locate the screen scheme in which you are interested, and click the **Configure** link next to it. The Configure Screen Scheme page is displayed.
4. Click the **Delete** link next to the issue operation you wish to remove.

### Activating a screen scheme

To activate a screen scheme, you need to associate it with one or more projects and issue types, using issue type screen schemes.

1. Configure an issue type screen scheme to use the screen scheme.
2. Associate the issue type screen scheme with a project.

For details of both procedures, see [Associating screen and issue operation mappings with an issue type](#).

# Associating screen and issue operation mappings with an issue type

What is an 'issue type screen scheme'?

An 'issue type screen scheme' associates a [screen scheme](#) (which defines mappings between screens and issue operations) with [issue types](#). Hence, an issue type screen scheme allows you to specify different [screens](#) for different issues types when used *for the same* issue operation (e.g. 'Create Issue') in a given Jira project. For more information, please see the [overview diagram](#).

By default, your Jira system contains an issue type screen scheme that's called a **default issue type screen scheme**. You may want to edit this scheme or copy it to make a new one.

## Configuring an issue type screen scheme

The configuration of an issue type screen scheme involves associating an issue type(s) with a particular screen scheme. For example, associating the 'Bug' issue type with the 'Default Screen Scheme', and then associating the 'Improvement' issue type with the 'Improvement Screen Scheme'.

**Note:** For all of the following procedures, you must be logged in as a user with the [Jira Administrators global permission](#).

### On this page:

- [What is an 'issue type screen scheme'?](#)
- [Configuring an issue type screen scheme](#)
- [Adding an issue type screen scheme](#)
- [Editing an issue type screen scheme](#)
- [Deleting an issue type screen scheme](#)
- [Copying an issue type screen scheme](#)
- [Associating an issue type screen scheme with a project](#)

## Associating an issue type with a screen scheme

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Screens > Issue Type Screen Schemes** to open the View Issue Type Screen Schemes page.
3. Click the **Configure** link next to the desired issue type screen scheme.
4. Click **Associate an issue Type with a Screen Scheme** and select the following options:
  - a. Select an **Issue Type** you wish to associate a screen scheme with.
  - b. Select the desired **Screen Scheme**.
5. Click the **Add** button and the new association will be added to the association list above.

### Please note

- There can only be one association for each issue type. If all issue types have been associated with a screen scheme, you can use the **Edit** link next to each entry to change the screen scheme that is associated with it.
- If there is no specific entry for an issue type, the screen scheme associated with the *Default* entry will be used.

## Editing an association

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Screens > Issue Type Screen Schemes** to open the View Issue Type Screen Schemes page.
3. Click the **Configure** link next to the desired issue type screen scheme, which opens the **Configure Issue Type Screen Scheme** page (see [above](#)).
4. Click the **Edit** link next to the issue type you wish to edit, which displays the **Edit Issue Type Screen Scheme Entry** page.
5. Select the screen whose association you wish to change, and click the **Update** button.

## Deleting an association

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Screens > Issue Type Screen Schemes** to open the View Issue Type Screen Schemes page.
3. Click the **Configure** link next to the desired issue type screen scheme, which opens the **Configure Issue Type Screen Scheme** page (see [above](#)).
4. Click the **Delete** link next to the issue operation you wish to remove.

**i** The *Default* entry is used for all issue types that do not have a specific entry in the scheme. It cannot be deleted.

## Adding an issue type screen scheme

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Screens > Issue Type Screen Schemes** to open the View Issue Type Screen Schemes page.
3. Click the **Add Issue Type Screen Scheme** button.
4. Enter the name for the new scheme. You can optionally add a description.
5. Select a screen scheme for the *Default* entry in the new scheme. The *Default* entry will be used for issue types that do not have a specific mapping in the scheme.
6. Click the **Add** button. The screen will automatically update the Issue Type Screen Schemes list with the new issue type screen scheme.

## Editing an issue type screen scheme

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Screens > Issue Type Screen Schemes** to open the View Issue Type Screen Schemes page.
3. Click the **Edit** link next to the desired issue type screen scheme to open the **Edit Issue Type Screen Scheme** page, where you can edit the issue type screen scheme's name and description, as well as the screen scheme of the *Default* entry.
4. Click the **Update** button, which returns you to the View Issue Type Screen Schemes page, with your updates now applied to the Issue Type Screen Schemes list.

## Deleting an issue type screen scheme

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Screens > Issue Type Screen Schemes** to open the View Issue Type Screen Schemes page.
3. Click the **Delete** link next to the issue type screen scheme you wish to delete.

**i** Issue type screen schemes that are associated with a project cannot be deleted.

## Copying an issue type screen scheme

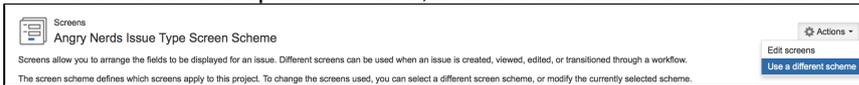
1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Screens > Issue Type Screen Schemes** to open the View Issue Type Screen Schemes page.
3. Click the **Copy** link next to the field screen you wish to copy, which opens the Copy Issue Type Screen Scheme page.
4. Enter the name and description of the new issue type screen scheme, and click the **Copy** button.

## Associating an issue type screen scheme with a project

Once you have created and configured an issue type screen scheme to your desired settings, you can now associate the scheme with a project. This will apply your chosen screen scheme to each issue type within the selected project.

1. Choose **Administration** (⚙️) > **Projects**, and select the relevant project.
2. Select the project you wish to configure by clicking on its name.
3. Select **Screens**.

4. Click the **Actions** drop-down menu, and choose **Use a different scheme**:



5. Select the screen scheme you wish to associate with this project.
6. Click the **Associate** button.

**i** To control which issue types apply to a project, please see '[Associating issue types with projects](#)'.

# Creating a notification scheme

Jira applications can generate **email notifications** for various *events* that happen throughout the lifecycle of an issue, including *custom events*. Notifications are defined within a *notification scheme* (see below), which associates particular events with particular email recipients. The notification scheme is then assigned to a particular [project](#).

 You can use the same notification scheme for more than one project.

**Note:** For all of the following procedures, you must be logged in as a user with the **Jira Administrators** [global permission](#).

## Default notification scheme

Jira applications are pre-packaged with a notification scheme called **Default Notification Scheme**:

- *This scheme is associated with all new projects by default.* This means that if you have an [outgoing \(SMTP\) mail server](#) set up, that email notifications will be sent as soon as there is any activity (e.g. issues created) in the new project.
- You can disassociate this notification scheme from the project via the **Project Summary** page, as described [below](#).
- You can modify this scheme, or create other notification schemes for particular projects.

## Creating a notification scheme

1. Choose **Administration**  > **Issues**.
2. Select **Notification Schemes** to open the Notification Schemes page, which lists all the current notification schemes in your Jira installation.
3. Start creating the new notification scheme, by doing either of the following:
  - Click the **Copy** link to copy an existing notification scheme. If you have a notification scheme whose event recipients are reasonably similar to what you require, creating a copy is the quickest way to add a new scheme.
  - OR
  - Click the **Add Notification Scheme** button. On the **Add Notification Scheme** page, enter a name for the notification scheme and a short description of the scheme.
4. If you added a new notification scheme or you copied an existing one but have clicked the **Edit** link to modify the automatically generated name and/or description of the copied notification scheme:
  - a. Enter a name (or modify the existing one) for the notification scheme (e.g. 'Angry Nerds Notification scheme').
  - b. (*Optional*) Enter a description (or modify the existing one) for the notification scheme.
  - c. Click the **Add** button to create the notification scheme.
5. Add notifications/recipients as described [below](#).
6. Associate your new notification scheme with a project as described [below](#).

## Adding an event recipient to a notification scheme

To add a new recipient for a particular event:

1. Choose **Administration**  > **Issues**.
2. Select **Notification Schemes**.
3. Find your notification scheme, and click its linked name to open the **Edit Notifications** page. The **Edit Notifications** page lists all of the [events](#), along with the recipients who will receive notifications when each event occurs.
4. Click **Add** in the appropriate event row (see the list of [events](#) below), which opens the **Add Notification** page, where you can choose who to notify (about the event) from the list of available [recipients](#) (see below).
5. Select the appropriate recipient (filling in any required information for your particular choice of recipient).
6. Click **Add**. You are taken back to the **Edit Notifications** page (see [above](#)), with the notification you just specified now listed against the appropriate issue event.
7. If you make a mistake, or you would like to remove who is being notified, simply click the **Delete** link beside the person/group/role.

## Associating a notification scheme with a project

1. Choose **Administration**  > **Projects**, and select the relevant project.
2. At the lower-right of the **Project Summary** page, locate the **Notifications** section, click the name of the current scheme (e.g. **Default Notification Scheme**) or **None** (if the project is not yet associated with a scheme) to display details of the project's current notification scheme.
3. Click the **Actions** dropdown menu and choose **Use a different scheme** (or **Select a scheme**).
4. On the subsequent **Associate Notification Scheme to Project** page, which lists all available notification schemes, select the notification scheme you want to associate with the project and click the **Associate** button.

### Events

Jira applications support the following events, which can generate email notifications (as defined in a notification scheme).

 Each event will be sent in a separate email, but you can enable [batching of email notifications](#) to get similar events grouped in a single summary email.

Event	Description
Issue created	An issue has been entered into the system.
Issue updated	An issue has had its details changed. This includes the deletion of an issue comment.
Issue assigned	An issue has been assigned to a new user.
Issue resolved	An issue has been resolved (usually after being worked on and fixed).
Issue closed	An issue has been closed. (Note that an issue may be closed without being resolved).
Issue commented	An issue has had a comment added to it.
Issue comment edited	An issue's comment has been modified.
Issue reopened	An issue has been re-opened.
Issue deleted	An issue has been deleted.
Issue moved	An issue has been moved into or out of this project.
Work logged on issue	An issue has had hours logged against it (i.e. a worklog has been added).
Work started on issue	The Assignee has started working on an issue.
Work stopped on issue	The Assignee has stopped working on an issue.
Issue worklog updated	An entry in an issue's worklog has been modified.
Issue worklog deleted	An entry in an issue's worklog has been deleted.
Generic event	The exact nature of this event depends on the <a href="#">workflow transition(s)</a> from it was fired.

<b>Custom event(s):</b>	The exact nature of these events depends on the <a href="#">workflow transition(s)</a> from which they were fired.
-------------------------	--

**i** Jira applications do not have a specific notification event for the deletion of issue comments. When an issue's comment is deleted, Jira sends out an email notification as an 'Issue Updated' event.

### Recipients

The following types of recipients can receive email notifications.

Recipient	Description
<b>Current assignee</b>	The user to whom the issue is currently assigned.
<b>Reporter</b>	The user who originally created the issue.
<b>Current user</b>	The user who performed the action that has triggered this event.
<b>Project lead</b>	The user who is managing the project to which the issue belongs.
<b>Component lead</b>	The user who is managing the component to which the issue belongs.
<b>Single user</b>	A particular user in your Jira system.
<b>Group</b>	A particular group in your Jira system.
<b>Project role</b>	The members of a particular project role for this project. <b>i</b> Note that it is recommended to use project roles (rather than groups) in your notifications as this can help minimize the number of notification schemes in your system.
<b>Single email address</b>	Any email address that you wish to alert. <b>i</b> A Single Email Address notification will only be sent if the issue is publicly viewable (as the email address of a non-Jira user could be specified, in which case a security check is not possible). Publicly viewable issues are issues which have a Permission scheme that gives the 'Browse Projects' permission to 'Anyone' (any non-logged-in users). The text template is used for notifications to a single email address.
<b>All watchers</b>	All users who are watching the issue.
<b>User custom field value</b>	The value of a custom field of type <i>User Picker</i> or <i>Multi User Picker</i> that may have been associated with issues. <b>i</b> An example of where this can be useful: if you have a custom User field called Tester, you can have the tester notified when an issue is resolved.
<b>Group custom field value</b>	The value of a custom field of type <i>Group Picker</i> or <i>Multi Group Picker</i> that may have been associated with issues..

### **i** Please note:

- Email notifications will only be sent to people who have permission to view the relevant issue that is, people who:
  - have the **Browse Projects** [project permission](#) for the project to which the issue belongs; and
  - are members of any [issue security levels](#) that have been applied to the issue.
- Jira can only send email notifications if SMTP email has been enabled (see [Configuring email notifications](#)).

- Jira's default setting is to not notify users of their own changes. This can be changed on a per user basis via their profile preferences.
- Each event is sent in a separate email, but you can enable [batching of email notifications](#) to get similar events grouped in a single summary email.

**⚠ Please also note:**

Jira will send notification emails to both the **previous assignee and the current assignee**, whenever the assignee field changes.

However, earlier versions of Jira only sent a notification email to the previous assignee *if* the operation that changed the event was the **Assign Issue** operation. It did not send a notification if the issue was edited in some other way.

The `jira.assignee.change.is.sent.to.both.parties` advanced Jira option allows this legacy behavior to be re-instated, for those customers who prefer this behavior.

See [JRA-6344](#) for more details.

# Using the issue collector

## What is an 'issue collector'?

The issue collector allows you to easily embed a Jira feedback form into your own web site. This form is typically accessed by clicking a 'trigger' tab exposed along the edge of pages in your web site.

When used by people visiting your web site click this trigger tab and submit the resulting Jira feedback form, an issue is conveniently created in Jira.

Visitors to your web site do not require a user account in Jira to use the Jira feedback form.

**Note:** For all of the following procedures, you must be logged in as a user with the **Jira Administrators** [global permission](#).

## Accessing Jira's issue collectors

In Jira applications, issue collectors are configured (and hence organized) on a per-project basis.

### To access all issue collectors configured in Jira:

1. Choose **Administration** (⚙️) > **System**.
2. Select **Issue Collectors** to open the Issue Collectors page, which shows a list of all existing issue collectors in your Jira system.
3. Click the name of a project to access a more detailed list of issue collectors belonging to that project or click the name of an issue collector to access detailed information about it. On the issue collector page (containing detailed information), you can access:
  - An activity graph, showing the number of issues created via this issue collector (Y-axis) on a daily basis (X-axis).
  - A list of recent issues in reverse chronological order, which have been created via this issue collector.

### To access issue collectors belonging to a specific project:

1. Choose **Administration** (⚙️) > **Projects**, and select the relevant project.
2. Select a project.
3. From the project administration page, click the **Issue Collectors** tab. The Issue Collectors page is displayed, listing any issue collectors that have already been set up in your project:

Name	Last Updated By	Issue Type	Description	Activity	Operations
Test issue collector	admin admin [Administrator]	Bug			Edit · Copy · Disable · Delete

4. Click the name of an issue collector to access detailed information about it in particular, its recent activity and details on how to embed the issue collector into your web site.

## Adding an issue collector

1. Choose **Administration** (⚙️) > **Projects**, and select the relevant project.
2. On the left of the Project Summary page, click the **Issue Collectors** tab. The Issue Collectors page is displayed, listing any issue collectors that have already been set up in your project.
3. Click the **Add Issue Collector** button to open the Add Issue Collector page.
4. In the top section of the Add Issue Collector page, specify the following:

### On this page:

- [What is an 'issue collector'?](#)
- [Accessing Jira's issue collectors](#)
- [Adding an issue collector](#)
- [Embedding an issue collector into your web site](#)
- [Editing an issue collector](#)
- [Copying an issue collector](#)
- [Disabling or deleting an issue collector](#)
- [Known limitations](#)

### Related pages:

- [Advanced use of the Jira issue collector](#)

<b>Name</b>	Specify the name of the issue collector, as you want it to appear throughout the Jira user interface.
<b>Description</b>	Specify a description for the issue collector. This description will appear adjacent to the name of your issue collector, throughout the Jira user interface.
<b>Issue type</b>	Select the type of issue that you want created in Jira when visitors to your web site submit your issue collector's Jira feedback form.
<b>Issue reporter</b>	Specify the username that will be the default reporter of Jira issues created when visitors to your web site submit your issue collector's Jira feedback form.
<b>Match reporter?</b>	Select either of the following: <ul style="list-style-type: none"> <li>• <b>Always use Issue Reporter</b> select this option to ensure that the default issue reporter you specify above will always be the reporter of issues created by submission of the Jira feedback form on your web site.</li> <li>• <b>Attempt to match submitter email address</b> select this option if you want the reporter of an issue created by submission of the Jira feedback form on your web site to be a Jira user whose email address matches the email address specified in the email field of the Jira feedback form.</li> <li>• <b>i</b> Please note that if the Jira user does not have the Create Issues <a href="#">project permission</a> in your Jira project, the default issue reporter you specify above will be used as the issue's reporter.</li> </ul>
<b>Collect browser info</b>	Select this option to collect meta-information about your browser's statistics, which will be incorporated into issues created by submission of the Jira feedback form on your web site.

5. In the middle section of the Add Issue Collector page (entitled 'Trigger'), specify the following:

<b>Trigger text</b>	Specify a short, brief phrase that will appear on the trigger tab on your web site.
<b>Trigger style</b>	Choose the style in which the trigger tab will appear on your web site. 'Custom' will not display a trigger, but will add additional javascript to the generated script, so you can create a custom trigger on your web page.

6. In the lower section of the Add Issue Collector page (entitled 'Issue Collector Form'), specify the following:

<b>Template</b>	<p>Choose from the options provided. Typically, your choice would reflect the type of issue being created (i.e. chosen above). You can choose:</p> <ul style="list-style-type: none"> <li>• A predefined template for your Jira feedback form either 'Got Feedback?' or 'Raise a Bug'.</li> <li>• <b>Custom</b> to create a custom Jira feedback form, which allows you to specify your own wording on the dialog box, as well as add or remove other fields on the form, and change their positions on the form. <ul style="list-style-type: none"> <li>• Please note that if a field is <i>required</i> for the chosen issue type but that field has: <ul style="list-style-type: none"> <li>• No specified a default value, the field will automatically appear on the form. This field's position can be changed on the form, although it cannot be removed.</li> <li>• A default value but the field is not added to the form, then the field's default value is used when an issue is created via the issue collector.</li> </ul> </li> <li>• Not all fields of types of fields can be added to the form, since some fields cannot be displayed to anonymous users. The fields types that can be displayed are: <ul style="list-style-type: none"> <li>• <i>Standard Fields</i>: Summary, Description, Components, AffectsVersion, Environment, Priority, Attachment</li> <li>• <i>Custom Field Types</i>: Date Time, Radio Buttons, Multi-Checkbox, Multi-Select, Number, Select List, URL field, Version Picker, Cascading Select, Project Picker, Single Version Picker, Text Field, Free Text Field</li> </ul> </li> </ul> </li> </ul>
<b>Message</b>	Type a message, which appears in the blue 'information' panel along the top of the dialog box.

7. Click the **Submit** button to save your changes.

## Embedding an issue collector into your web site

After clicking the **Submit** button to save your new issue collector, a page containing code snippets is displayed. Use the code and information provided to embed your new issue collector into your web site.

✔ If you accidentally click away from this page, you can easily retrieve the information that was on it by accessing your issue collector's details ( [above](#) ) and scrolling to the end of the page.

## Editing an issue collector

Editing an issue collector should not require any changes to web pages that include the issue collector, unless you change the trigger style to or from a custom trigger. Changing the trigger style to or from a custom trigger will change the generated javascript, so you may need to change what you embed in any web page that includes the issue collector.

1. Log in to Jira as a [project administrator](#) or a user with the **Jira Administrators global permission**.
2. Access the relevant project's list of issue collectors ( [above](#) ).
3. In the Operation drop-down for the issue collector you would like to edit, select **Edit** to open the Edit Issue Collector page.
4. Update the issue collector, as desired.
5. Click **Update** to save your changes.

## Copying an issue collector

Copying an issue collector will create an entirely new issue collector and will not affect any existing issue collectors. You will need to embed it in whatever web pages you would like, just as if you had created a new issue collector.

1. Log in to Jira as a [project administrator](#) or a user with the **Jira Administrators global permission**.
2. Access the relevant project's list of issue collectors ( [above](#) ).
3. In the Operation drop-down for the issue collector you would like to copy, select **Copy** to open the Add Issue Collector page.
4. All the information from the copied issue collector will be the same as the copied issue collector, with the exception of the name (which will be "Copy of " + *the original name of the copied issue collector*).
5. Update the issue collector, as desired.
6. Click **Submit** to save your changes

## Disabling or deleting an issue collector

1. Access the relevant project's list of issue collectors ([above](#)).
2. On the list of the project's issue collectors, click **Disable** or **Delete** to respectively disable or delete the associated issue collector.
  - ⓘ While an issue collector is disabled, its trigger tabs will still be visible on pages of your web site(s) to which the issue collector code has been added until a user refreshes the page. However, clicking these triggers results in a message indicating that the issue collector is currently out of action.

## Known limitations

- To improve security, project and issue keys are no longer displayed in the success message after submitting the feedback (unless the project is open to Anyone on the web). If you need to display them, you can use [this workaround](#).
- Placing the Issue Collector plugin within an iframe will not close the prompt window automatically.

This is a known limitation for the Issue Collector plugin, and has been tracked at

🔴 **JRASERVER-41400** - Issue Collector Cannot Be Closed When Placed Inside an iframe  
GATHERING IMPACT

- If an anonymous user tries to create an issue collector in a project and there are required fields present in this project, these fields are not shown to the anonymous user. As a result, the user cannot create the collector.  
The workaround is to make the fields optional.

This is a known limitation for the Issue Collector plugin, and has been tracked at

🔴 **JRASERVER-67094** - Unable to create Issue Collectors, because of required fields not visible to anonymous users  
GATHERING IMPACT

# Advanced use of the Jira issue collector

## Customizing the Jira issue collector

The Jira issue collector can be used without any additional JavaScript beyond the single line generated in the issue collector administration screens in Jira. However, you can also customize the Jira issue collector in a number of different ways:

- Set up a custom trigger, so the feedback form launches from a different link or button than the packaged triggers provided.
- Set the default values of fields for your users, using JavaScript.
- Specify the values of fields on the issue, which are not shown in the feedback form.

This page assumes you are already familiar with [using the issue collector](#).

**⚠ Warning:** The JavaScript exposed by the issue collector is not considered a stable API and may change with new Jira releases.

### On this page:

- [Customizing the Jira issue collector](#)
- [Setting up a custom trigger](#)
- [Adding the custom trigger function manually](#)
- [Setting field values from JavaScript](#)
- [Embedding multiple issue collectors](#)
- [Embedding the issue collector](#)

## Setting up a custom trigger

### Configuring your collector to use a custom trigger

If you want to use a different trigger, or button, to launch the issue collector on your website, configure your issue collector as described below:

1. Add a new issue collector, or edit an existing issue collector.
2. Scroll down to section **Trigger** and select the option 'Custom'.
3. You don't need to set any **Trigger Text** as this will be overridden by your custom trigger.

### Adding the issue collector script for a custom trigger

**⚠** Creating and debugging custom scripts are outside of the scope of Atlassian Support. For assistance, please post any questions at <https://answers.atlassian.com>

The issue collector script generated by Jira for adding a custom trigger is slightly different to the script generated for the standard triggers, because it includes the JavaScript function for the custom trigger.

Customization of the issue collector is done by creating/extending the global object **ATL\_JQ\_PAGE\_PROPS**. This allows you to add a custom trigger, set default values for fields and more.

**Note:** In Jira 5.1 (and version 1.1 of the Issue Collector plugin), the issue collector administrative interface let you define the custom trigger function UI, and you did not need to include it in the JavaScript on the page. In version 1.2 of the Issue Collector, the custom trigger JavaScript is a part of the generated JavaScript that you should copy and paste into your web page.

The code snippet below shows a sample HTML page with the generated issue collector JavaScript.

In the example below, we've added a simple button in HTML, and made that button launch the issue collector. This is done simply by replacing 'myCustomTrigger' in the generated JavaScript with the HTML id of the button ('feedback-button')

```

<head>
  <!-- We pasted the generated code from the
  Issue Collector here, after choosing a custom
  trigger -->

  <!-- This is the script for the issue
  collector feedback form -->

  <script type="text/javascript" src="<Jira
  URL>/s/en_US-ydn9lh-418945332/803/1088/1.2/_
  /download/batch/com.atlassian.jira.collector.
  plugin.jira-issue-collector-plugin:issuecollector
  /com.atlassian.jira.collector.plugin.jira-issue-
  collector-plugin:issuecollector.js?
  collectorId=d03d7bd1"></script>

  <!-- This is the script for specifying the
  custom trigger. We've replaced 'myCustomTrigger'
  with 'feedback-button' -->

  <script type="text/javascript">
    window.ATL_JQ_PAGE_PROPS = {
      "triggerFunction": function
(showCollectorDialog) {
          //Requires that
          jQuery is available!
          jQuery("#feedback-
          button").click(function(e) {
              e.
              preventDefault();
              showCollectorDialog();
          });
        };
      };
    };
  </script>

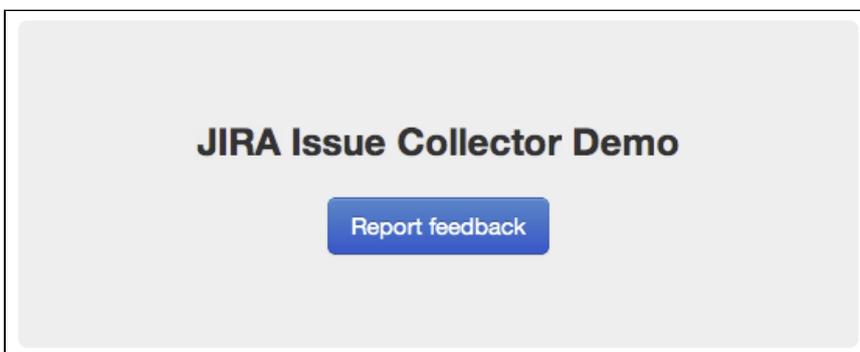
</head>

<body>

  <h2>Jira Issue Collector Demo</h2>
  <a href="#" id="feedback-button"
  class='btn btn-primary btn-large'>Report feedback<
  /a>

</body>

```



### Adding the custom trigger function manually

*The custom trigger JavaScript will be included in the JavaScript generated by the issue collector. However, this section provides details on how you could do it without pasting in the additional lines of generated JavaScript.*

To add a custom trigger, add the property **triggerFunction** in the global object **ATL\_JQ\_PAGE\_PROPS**. **triggerFunction** needs to be defined as a function and takes one argument which is the function for displaying the issue collector.

You can invoke the issue collector from any element on your page by adding a click handler in **triggerFunction** as shown below. In this example, we will be calling the issue collector from our **#feedback-button** anchor tag defined in the above HTML markup. You can assign multiple triggers for the same issue collector by adding more click handlers.

```

window.ATL_JQ_PAGE_PROPS = $.extend(window.ATL_JQ_PAGE_PROPS, {

    // ==== custom trigger function ====
    triggerFunction : function( showCollectorDialog ) {
        $('#feedback-button').on( 'click', function(e) {
            e.preventDefault();
            showCollectorDialog();
        });

        // add any other custom triggers for the issue collector here
    }

});

```

The **triggerFunction** will be invoked by the issue collector after the `$(document).ready()` phase.

## Setting field values from JavaScript

### Setting field values

The issue collector gives you the option to set field values for any of the fields on the issue type. This is done by adding the property **fieldValues** in the global object **ATL\_JQ\_PAGE\_PROPS**. There are different methods for setting default values for different field types. The code samples below show a visual representation of a field in Jira and its relevant markup, and how to set a default value for that field type. Use a DOM inspection tool such as Firebug in the Jira Issue Create Screen to extract the field names and values relevant to your issue collector. Please note that the Issue Collector is not supposed to be a replacement for the [Jira REST API](#). If you require a more customized solution, make use of the Jira REST API to create Jira issues from external websites. The [Jira Travel App](#) is a good example of how you can build a front end interface with Jira as the back end.

### Visible fields (setting default field values)

If you set the value of a field that is visible on the issue collector feedback form, the fields will already be filled in with that value when the form opens.

### Hidden fields

There might be cases where you might want to set a field value without actually displaying the field on the issue collector. In this case, simply use the same method as above to set the field values via JavaScript. The fields will not be shown as they were not added in the form template but their values will still be present in issues created with the issue collector.

### JavaScript for setting field values

Setting field values is done by specifying field name / value pairs within the "fieldValues" block of `window.ATL_JQ_PAGE_PROPS`. If you already have a custom trigger defined, you can simply add to the definition of `window.ATL_JQ_PAGE_PROPS` like the example below.

Note the names of the fields are always the names of the field in the Jira Create Issue Screen, not any overridden names you may have provided in the issue collector form.

```

window.ATL_JQ_PAGE_PROPS = $.extend(window.ATL_JQ_PAGE_PROPS, {

    // ==== custom trigger function ====
    triggerFunction : function( showCollectorDialog ) {
        $('#feedback-button').on( 'click', function(e) {
            e.preventDefault();
            showCollectorDialog();
        });
    },
    // ==== we add the code below to set the field values ====
    fieldValues: {
        summary : 'Feedback for new website designs',
        description : 'The font doesn\'t quite look right',
        priority : '2'
    }
});

```

## Examples of how to set specific field types

### Text field example

Setting the value for a text field, like the issue summary, is straightforward. Here's the markup for a text field like Summary in the issue collector (you do not need to add this, this is simply to show the representation that the issue collector contains):

```

<div class="field-group">
    ...
    <input class="text long-field" id="summary" name="summary" type="text" value="">
    ...
</div>

```

And here's how you set the value of the field in JavaScript:

```

fieldValues : {
    summary : 'This is the default summary value'
}

```

### Select list example with issue priority

Setting the value for a select list field, such as the issue priority, requires a little more effort, because you need to know the HTML element id for the choice you want to select. Here's the markup for the Priority field in the issue collector (you do not need to add this, this is simply to show the representation that the issue collector contains):

```

<div class="field-group">
    ...
    <input id="priority-field" class="text aui-ss-field ajs-dirty-warning-exempt" autocomplete="off">
    ...
    <select class="select" id="priority" name="priority" style="display: none; " multiple="multiple">
        <option class="imagebacked" data-icon="/images/icons/priority_blocker.gif" value="1"
    >Blocker</option>
        <option class="imagebacked" data-icon="/images/icons/priority_critical.gif" value="2"
    >Critical</option>
        ...
    </select>
    ...
</div>

```

And here's how you set the value of the field in JavaScript:

```
fieldValues : {
  'priority' : '2'
}
```

### Multi-select or checkboxes example

Setting the value for a multi-select (like the Browser field) or checkbox requires that you provide an array of values. Like the select list, you need to know the values to set, by looking at the markup on the Create Issue Screen.

```
<div class="field-group">
  ...
  <select class="select" id="customfield_10110" multiple="multiple" name="customfield_10110" size="
5">
    <option value="-1" selected="selected">None</option>
    <option value="10039">All Browsers</option>
    <option value="10037">Chrome</option>
    ...
  </select>
  ...
</div>
```

And here's how you set the value of the field in JavaScript: the field values must be set as an array of values, even if there is only one value.

```
fieldValues : {
  'customfield_10110' : [ '10039', '10037' ]
}
```

### Custom fields

Setting a value for a custom field is exactly the same as any other field in Jira. Since multiple custom fields can share the same name, custom fields will be referenced by "customfield\_" + the Id of the custom field in Jira. This ID can be seen in the HTML markup for the Create Issue Screen in Jira, but can also be determined by looking at the URLs on the custom fields screen in Jira administration. Here's what the JavaScript would look like for setting a custom field whose id in Jira was 11111:

```
fieldValues : {
  'customfield_11111' : 'San Francisco'
}
```

### Cascading selects

Setting a value for a cascading select is done in two steps - one for the parent value and one for the child. Below is an example of setting the value of a cascading select field.

```
fieldValues : {
  'customfield_12345' : 'Australia',
  'customfield_12345:1' : 'Sydney'
}
```

## Special case fields

### Environment field

By default, the issue collector puts user context such as the URL, User-Agent and screen resolution in the environment field. There might be cases where you wish to include more information in the environment field. In this case, you can add the property `environment` in the global object `ATL_JQ_PAGE_PROPS`. This allows you to add key value pairs that will appear on the environment field in the Jira issue.

```

window.ATL_JQ_PAGE_PROPS = $.extend(window.ATL_JQ_PAGE_PROPS, {
  // ==== custom trigger function ====
  triggerFunction : function( showIssueCollector ) {
    ...
  },
  // ==== default field values ====
  fieldValues : {
    ...
  },
  // ==== Special field config for environment ====
  environment : {
    'Custom env variable' : $('#build-no').text(),
    'Another env variable' : '#007'
  }
});

```

### Restricted fields

Some fields that require a user to be logged into Jira cannot be set through JavaScript. Assignee is an example of a field that cannot be set via JavaScript.

### Dynamic functions

**Environment** and **fieldValues** properties can also be a function returning a JSON object that will be executed immediately when the collector trigger is shown (**not** just before opening the collector form). This might come in handy when you might wish to capture contextual information relevant to the user.

```

window.ATL_JQ_PAGE_PROPS = $.extend(window.ATL_JQ_PAGE_PROPS, {
  // ==== custom trigger function ====
  triggerFunction : function( showIssueCollector ) {
    ...
  }
  // ==== Special field config for environment ====
  , environment : function() {

    var env_info = {};

    if ( window.ADDITIONAL_CUSTOM_CONTEXT ) {
      env_info[ 'Additional Context Information' ] = window.ADDITIONAL_CUSTOM_CONTEXT;
    }

    return env_info;
  }
  // ==== default field values ====
  , fieldValues : function() {

    var values = {};

    var error_message = $('#error_message');
    if ( error_message.length !== 0 ) {
      // record error message from the page context rather than asking the user to
      enter it
      values[ 'summary' ] = error_message.children('.summary').text();
      values[ 'description' ] = error_message.children('.description').text();
    }

    return values;
  }
});

```

### Embedding multiple issue collectors

If you want to have two different forms appear on the same web page, you will need to create two different issue collectors in Jira. To set custom triggers, or set field values on those issue collectors requires a few changes to your page:

1. Include the generated JavaScript for both of your issue collectors in the page.
2. Find the id of each collector. This can be done one of two ways:

- a. The parameter of the script is "collectorId=<8 character id>. That's the ID you want.
- b. Go to the Issue Collector page in the Admin section and click on the Issue Collector you wish to embed. Copy the collectorId from the URL.

```
https://<Jira_URL>/secure/ViewCollector!default.jsps?projectKey=<PROJECT_KEY>&collectorId=<copy this part here>
```

Then, create separate namespaces for each of the issue collectors in the `ATL_JQ_PAGE_PROPS` object.

```
window.ATL_JQ_PAGE_PROPS = $.extend(window.ATL_JQ_PAGE_PROPS, {
  '<collectorId_1>' : {
    triggerFunction:
      // define trigger function

    fieldValues: {
      // define field values here
    }
  },
  '<collectorId_2>' : {
    triggerFunction:
      // define trigger function
    fieldValues: {
      //define field values here
    }
  }
});
```

## Embedding the issue collector

### Embedding the issue collector in your Confluence Site

The issue collector can be embedded into Confluence using the [HTML Macro](#). Note that using the HTML Macro would require you to embed the issue collector code separately on each page.

The issue collector was previously embeddable in Confluence via a [User Macro](#), allowing you to create a re-usable issue collector macro that other Confluence users can embed into their pages. This option is currently unavailable due to a known bug:

 **CONFSERVER-26104** - Some JavaScripts are not executed if included in User Macro

GATHERING IMPACT

Embedding the issue collector is not currently supported in Confluence Cloud.

### Jira

The issue collector can be embedded in the announcement banner on a Jira page by embedding the above script and HTML markup for your custom trigger in the [announcement banner configuration screen](#). If you wish to change the location of your custom trigger, this can be easily done via jQuery. The following snippet shows how you can add the custom trigger onto the footer of all Jira pages.

You cannot embed an issue collector in your Jira Cloud site since HTML markup is disabled for the announcement banner.

```

window.ATL_JQ_PAGE_PROPS = $.extend(window.ATL_JQ_PAGE_PROPS, {
  // ==== custom trigger function ====
  triggerFunction : function( showIssueCollector ) {
    // button markup - relevant css can be added via the style attribute
    var feedbackButton = "<a id='feedback-button'>Got Feedback?</a>";
    // embed the button in the footer
    $('#footer-link').append(feedbackButton);

    $('#feedback-button').click(function(e) {
      ...
    });
  }
});

```

Please note that embedding the issue collector requires you to enable HTML markup for the announcement banner.

### Full source code

This source code shows how to embed two different issue collectors on the same page with custom triggers.

```

<body>

  <h2>Jira Issue Collector Demo</h2>
  <a href="#" id="feedback_button" class='btn btn-primary btn-large'>Report feedback</a><br />
  <a href="#" id="bug_button" class='btn btn-primary btn-large'>Report bug</a>

  <!-- Jira Issue Collector - append this at the bottom of <body> -->
  <script type="text/javascript" src="https://<Jira URL>/s/en_US-ydn9lh-418945332/803/1088/1.2/_
/download/batch/com.atlassian.jira.collector.plugin.jira-issue-collector-plugin:issuecollector/com.
atlassian.jira.collector.plugin.jira-issue-collector-plugin:issuecollector.js?
collectorId=<collectorId_1>"></script>
  <script type="text/javascript" src="https://<Jira URL>/s/en_US-ydn9lh-418945332/803/1088/1.2/_
/download/batch/com.atlassian.jira.collector.plugin.jira-issue-collector-plugin:issuecollector/com.
atlassian.jira.collector.plugin.jira-issue-collector-plugin:issuecollector.js?
collectorId=<collectorId_2>"></script>

  <!-- We will customize Jira in the following script tag -->

  <script type="text/javascript">
    // safely use jquery here since the issue collector will load it for you
    $(document).ready(function() {

      window.ATL_JQ_PAGE_PROPS = $.extend(window.ATL_JQ_PAGE_PROPS, {

        // ==== feedback collector ====
        '<collectorId_1>' : {

          // === custom trigger function ===
          triggerFunction : function( showCollectorDialog ) {
            $('#feedback_button').click( function(e) {
              e.preventDefault();
              showCollectorDialog();
            });
          }

          // === default and hidden field values ===
          , fieldValues : {

            // default values
            summary : 'Feedback for new website designs'
            , description : 'The font doesn\'t quite look right'

            // hidden field value
            , priority : '2'

          }

        }

      });

      // ==== bug collector ====

```

```

        , '<collectorId_2>' : {
            // === custom trigger function ===

            triggerFunction : function( showCollectorDialog ) {
                $('#bug_button').click( function(e) {
                    e.preventDefault();
                    showCollectorDialog();
                });
            }

            // === additional environment details ===
            , environment : function() {

                var env_info = {};

                if ( window.ADDITIONAL_CUSTOM_CONTEXT ) {
                    env_info[ 'Additional Context Information' ] =
window.ADDITIONAL_CUSTOM_CONTEXT;
                }

                return env_info;
            }
            // === default field values ===
            , fieldValues : function() {

                var values = {};

                var error_message = $('<error_message>');
                if ( error_message.length !== 0 ) {

                    // record error message from the page context
                    values[ 'summary' ] = error_message.children('<summary>').text();
                    values[ 'description' ] = error_message.children('<description>').text();
                }

                return values;
            }
        }
    });
</script>
</body>

```

### Is localization of an issue collector possible?

You can create an issue collector 100% localized to the default language of your Jira instance. Beyond that, complete localization of the issue collector is not possible.

The strings and text in the issue collector feedback form of the issue collector is a combination of:

1. The issue collector strings set by the Jira Administrator
2. Either the default language setting for Jira, or the language preference of the user if they are logged in to Jira.
  - All users will see the names of the fields as they are set by the Jira Administrator. These are not affected by the default language of Jira, and are not affected by the default language of logged in Jira users.
  - All users will see the field descriptions as they are set in the Jira Administration UI.
  - For everything else:
    - **Anonymous** users will see everything else in the default Jira language.

- Logged in users will see everything else in the feedback form in the language specified by their Jira profile.

Because of the above, you cannot create a single issue collector that will present itself entirely in the language of the end user.

However, if you want to create an issue collector that will present itself to anonymous users in the default language of your Jira instance, you should:

1. Use the custom feedback template for the issue collector
2. Change the field labels in Jira, and the labels for name and email, to the words you want to use in the default Jira language.

The language setting of the browser will not impact the text in the feedback form.

# Working with workflows

A Jira workflow is a set of *statuses* and *transitions* that an issue moves through during its lifecycle and typically represents processes within your organization. There are default built-in workflows that cannot be edited; however, you can copy and use these workflows to create your own.

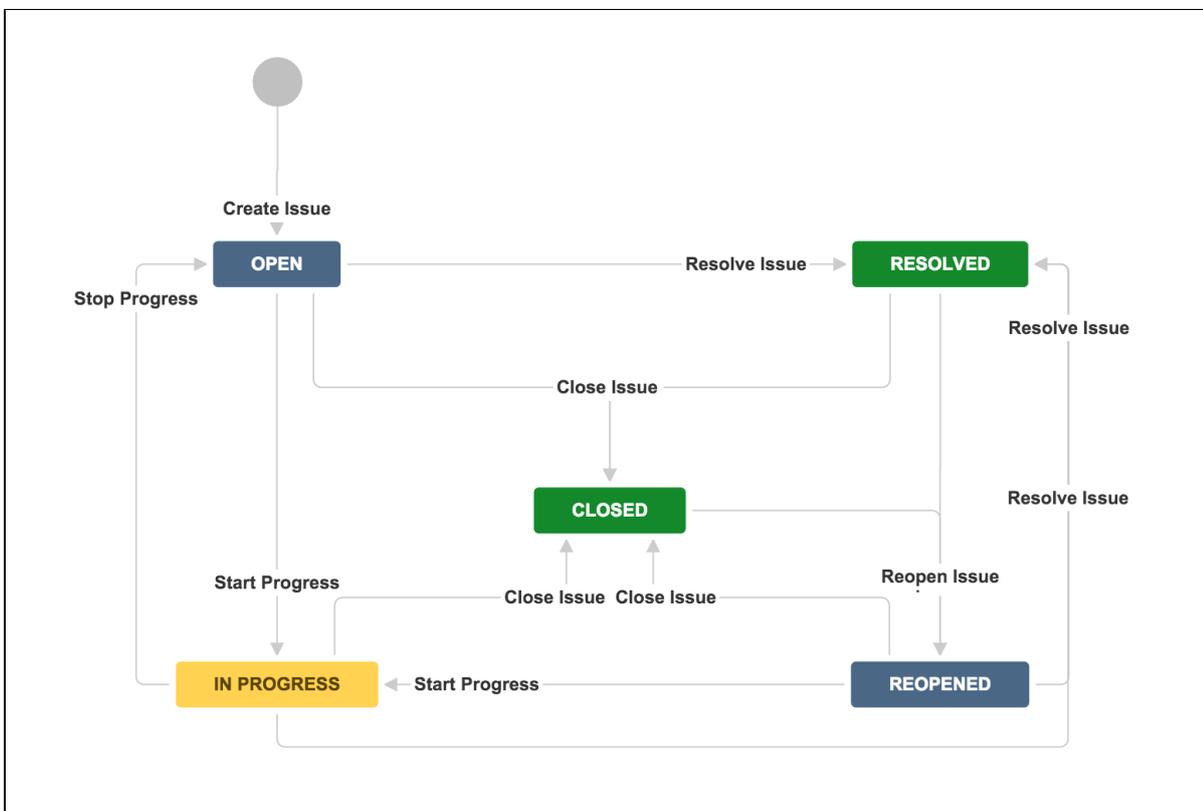
You can also create your own workflows from scratch, or import workflows from Atlassian Marketplace. Workflows can be associated with particular projects and, optionally, specific *issue types*, by using a [workflow scheme](#).

You will need to log in as a user with the 'Jira Administrators' [global permission](#) to access and manage workflows. If the workflow isn't the default workflow, or shared with another project, project administrators also have limited editing permission to the workflow.

**On this page:**

- [Statuses and transitions](#)
- [Active and inactive workflows](#)
- [Workflow designer](#)
- [Creating workflows](#)
- [Configuring a workflow](#)
- [Advanced workflow configuration](#)

Here's an example of a default workflow:



## Statuses and transitions

A status represents the state of an issue at a specific point in your workflow. An issue can be in only one status at a given point in time. When defining a status, you can optionally specify [properties](#).

A transition is a link between two statuses that enables an issue to move from one status to another. In order for an issue to move between two statuses, a transition must exist.

A transition is a *one-way* link, so if an issue needs to move back and forth between two statuses, two transitions need to be created. The available workflow transitions for an issue are listed on the View issue screen.

## Active and inactive workflows

There are slight differences between editing an inactive and an active workflow. We place restrictions on the modifications you can make to an active workflow, due to the impact the changes will have on projects and/or issue types that use this workflow.

Workflow status	Description
Inactive workflow	An <i>inactive workflow</i> is a workflow that is not currently being used by any projects. Because there are no issues currently transitioning through an inactive workflow, you can edit the workflow's steps and transitions directly. For details on this, see <a href="#">Working in text mode</a> .
Active workflow	<p>An <i>active workflow</i> is a workflow that is currently being used by one or more projects. When you edit an active workflow, Jira first creates a draft of it, that you can then modify as you see fit. When you've finished, you can publish your draft and, optionally, save your original workflow as an inactive backup.</p> <p>The following limitations apply when editing the draft for an active workflow:</p> <ul style="list-style-type: none"> <li>• It is not possible to edit the workflow name (only the description) if a workflow is active.</li> <li>• Workflow statuses cannot be deleted.</li> <li>• The step ID cannot be changed. See <a href="#">Cannot Add Transitions or Delete Steps in Draft Workflows</a>.</li> </ul> <p>To make any of the modifications listed above, you need to copy the workflow (see <a href="#">Creating a workflow</a>), modify the copy, and then <a href="#">activate</a> it.</p>

## Workflow designer

The workflow designer is a graphical tool that allows you to see the layout of your workflow and to create and edit a workflow's steps and transitions. You will need to log in as a user with the 'Jira System Administrators' [global permission](#) to access the functionality described below.

With the workflow designer, you can:

- Manage status and transitions: add, click and drag, or select to edit properties ([Workflow properties](#)) to rename, or delete (from the workflow but not Jira).
- Add a global transition that allows every other status in the workflow to transition to the selected status. Select **Allow all statuses to transition to this one** in the properties panel for the transition.
- Change the screen that a transition uses. See [Working in text mode](#) for details.
- Configure advanced transition options, such as triggers, conditions, validators, and post functions. See the [Advanced workflow configuration](#) page.
  - Statuses are *global objects*. Changing the name of a status on one workflow also changes it in *all workflows that use that status*.
  - Hover over a transition or a status to see the relevant transition labels.
  - Zoom the diagram with your mouse wheel. Pan the diagram by clicking and holding the mouse while on white space, then moving your mouse across the diagram.
  - You cannot clone transitions in the workflow designer.
  - You cannot create annotations in the workflow designer.
  - You cannot directly set the `issue.editable` property. To do this, simply add the `issue.editable` property to the [status properties](#).
  - The workflow designer will automatically validate your workflow and highlight any statuses that have no incoming or outgoing transitions. The workflow validator will also highlight all transitions that have an invalid permission condition that you don't have available in Jira. The validator is particularly useful if you import workflows, or deal with complex workflows.

## Creating workflows

There are a few ways you can start a new workflow. These include cloning an existing workflow, creating a new workflow, and importing a workflow.

## Copy an existing workflow

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Workflows** to open the Workflows page, which displays all of the workflows in your system.

Name	Last modified	Assigned Schemes	Steps	Operations
jira (Read-only System Workflow) <b>DEFAULT</b> The default JIRA workflow.		• Product Development Workflow Scheme	5	View · Copy
JIRA Service Desk IT Support Workflow generated for Project TD This JIRA Service Desk IT Support Workflow was generated for Project TD	09/Jan/15 System Administrator	• JIRA Service Desk IT Support Workflow Scheme generated for Project TD	3	View · Edit · Copy

3. Copy an existing workflow using the **Copy** link in the Operations column (shown above). Enter a name and description and select the **Copy** button.
4. Customize it by adding or editing steps and transitions.

When you have finished customizing your workflow, see [Managing your workflows](#) for details on how to use it with a Jira project.

## Create a new workflow

### For advanced administrators

1. Click **Workflows** in the left-hand nav panel, then **Add Workflow** at the top of the screen.
2. Enter a name and description for your workflow. Click **Add**.  
The workflow opens in edit mode, and contains a step called **Open** and an incoming transition called **Create**.
3. Continue with your workflow customizations, by adding and editing steps and transitions.

## Import a workflow

Please see the documentation on [Importing workflows](#).

## Configuring a workflow

### Editing a project's workflow

Whenever you create a new Jira project, your project automatically uses the default workflow scheme. The scheme associates all available issue types in the project with the Jira system workflow. Since neither the Jira system workflow nor the default workflow scheme are editable, Jira creates an editable copy of the system workflow and workflow scheme for your project.

1. Choose **Administration** (⚙️) > **Projects**, and select the relevant project.
2. On the Administration page for the project, click **Workflows**.
3. Click the 'edit' icon at the top-right of the box, and Jira automatically does the following:
  - Creates a draft copy of the system workflow named 'Your Project Name Workflow (Draft)'.
  - Creates a new workflow scheme for the workflow named 'Your Project Name Workflow Scheme'.
  - Associates any existing issues in your project with the new workflow.
4. You can now edit your draft workflow. Click on a status or transition to see editing options in the panel that appears.
5. When you are finished, click **Publish**. The dialog allows you to publish your draft and, optionally, save your original workflow as an inactive backup.

- The number of issues impacts the speed when configuring a workflow - for small numbers of issues, this process is relatively quick, however if you have many (e.g.thousandsof) existing issues in your Jira project, this process may take some time.
- Once this process begins,*it cannot be paused or canceled*. Please avoid editing or transitioning any issues within your project while this process is taking place.

### Setting the resolution field

In Jira, an issue is either open or closed, based on the value of its 'Resolution' field not its 'Status' field.

- An issue is open if its resolution field has not been set.
- An issue is closed if its resolution field has a value (e.g.Fixed, Cannot Reproduce).

This is true regardless of the current value of the issue's status field (Open, In Progress, etc). Therefore, if you need your workflow to force an issue to be open or closed, you will need to set the issue's resolution field during a transition. There are two ways to do this:

- Set the resolution field automatically via [a post function](#).
- Prompt the user to choose a resolution via a screen. See [Working in text mode](#) for details on this.

### Renaming workflow transition buttons

If you copied the system workflow and you wish to rename the workflow transition buttons on the View Issue page, you must delete the following properties from all transitions in the copied workflow:

- `jira.i18n.title`
- `jira.i18n.description`

Otherwise, the default names (i.e.values of these properties) will persist. Read more about [transition properties](#).

### Working in text mode

Text mode is an advanced way of working with workflows, and it shows the difference between steps and statuses. In text mode, you work directly with steps. For details, see [Working in text mode](#).

### Advanced workflow configuration

See the documentation on [Advanced workflow configuration](#).



#### Do more with Jira

Discover even more ways to configure workflows with [top workflow apps](#) on the Atlassian Marketplace.

# Managing your workflows

Workflows need to be activated to use them in Jira. Activating a workflow is the process of mapping the workflow to a workflow scheme, and then associating the workflow scheme with a project. To configure a workflow scheme, see [Configuring workflow schemes](#).

A workflow scheme defines a set of associations or mappings between a workflow and an issue type. Workflow schemes are associated with a project and make it possible to use a different workflow for every combination of project and issue type.

For all of the following procedures, you must be logged in as a user with the **Jira Administrators** [global permission](#).

## On this page:

- [Activating a workflow](#)
- [Managing workflows for projects](#)
- [Exporting your workflow](#)
- [Importing workflows](#)

## Activating a workflow

Active workflows are those that are currently being used, while inactive workflows are those that are not associated with any workflow schemes, or are associated with workflow schemes that are not associated with any projects. Active workflow schemes are also those associated with projects, while inactive workflow schemes are not.

1. Create a workflow scheme or find an existing workflow scheme. See [Configuring workflow schemes](#) for instructions.
2. Configure the workflow scheme to use your workflow. See [Configuring workflow schemes](#) for instructions.  
Associate your workflow scheme with a project, as described in the [Associating a workflow scheme with a project](#) section below.

## Managing workflows for projects

You can manage your workflows by associating workflow schemes, importing, exporting, uploading, and sharing.

### Associating a workflow scheme with a project

You can associate a single workflow scheme with more than one project, although only one workflow scheme can be associated with a given project. The [issue type scheme](#) associated with a project defines the issue types that are available to that project. If an issue type is not defined in the project's issue type scheme, its workflow is not used.

1. Choose **Administration** () > **Projects**, and select the relevant project. The Project Summary page is displayed.
2. Click **Workflows** on the left of the Project Summary page (you can also click the **More** link in the Workflows section in the middle of the screen). This is the current workflow scheme used by the project.
3. Click the **Switch Scheme** link to display the Associate Workflow Scheme to Project page.
4. Select the relevant workflow scheme from the Scheme list and click the **Associate** button to begin the migration process.  
Each issue has to be in a valid status. The valid statuses for an issue are defined by its workflow. This means that when changing a workflow, you may need to tell Jira the status for specific issues after the change.
5. A screen displays that indicates the progress of migrating all the project's issues to the updated scheme's workflows. **Acknowledge** to finish the process.

### Disassociating a workflow scheme from a project

A Jira project must always be associated with a workflow scheme, since all issues must move through a workflow, even if that workflow only consists of a single *Create Issue* transition. By default all Jira projects with unmodified workflows use Jira's system workflow. *Disassociating* a workflow scheme re-associates your project's workflow with Jira's default workflow scheme.

1. Follow the instructions in [Associating a workflow scheme with a project](#) above.
2. When selecting the workflow scheme from the Scheme list, select the **Default** workflow scheme
3. Click the **Associate** button, and follow the wizard, which guides you through migrating all of the project's issues.

## Exporting your workflow

The workflow sharing feature allows you to share your team's workflow with other teams in your organization on different Jira instances, or external parties in other organizations via the [Atlassian Marketplace](#). This feature allows you to easily share and use workflows that other people have published, or to move a workflow from staging to production in your own organization. If you wish to share your Jira Workflow with another instance of Jira or upload it to the Atlassian Marketplace, you first need to download it.

1. Choose **Administration**  > **Issues**.
2. Find the workflow you wish to share by clicking on the **Workflows** section in the left-hand panel.
3. Click **View** or **Edit** under the Operations column.
4. Select **Export > As Workflow** and click **Next** to continue.
5. In the **Add Notes** field, add any special configuration notes; for example, information about plugins that should be installed. Jira auto-populates these notes for you when it discards parts of your workflow (for example, plugins, post functions, conditions, validators).
6. Click **Export** and select a download location. Ensure the location is publicly accessible.

## Uploading to Atlassian Marketplace

To share your workflow with other Jira users, upload it to the Atlassian Marketplace.

1. Create an account on [Atlassian Marketplace](#), or log in and choose **Manage apps** (more info: [Step-by-step Paid-via-Atlassian Listing](#)).
2. Click **Create new app**.
3. Choose **My app is not directly installable** (ensure that 'App Type' is listed as 'Not a Plugin'). You will need to host the workflow on your own servers, and add information about where the workflow export can be accessed in the **Binary URL** textbox. This should be the location you specified in step 6 of the prior instruction set.
4. Fill out the submission form, be sure to note the following:
  - a. The **Summary** field contains the information that will be displayed to users searching the Marketplace.
  - b. The **Category** for your workflow must be **Workflow Bundles**. Choosing **Workflow Bundles** ensures other Jira users will have visibility to your workflow.
  - c. The **App Key** must be unique, as it uniquely identifies your application; it will become the **application URL**.

You don't have to complete the form in one session. You can save your form and come back to it later. Once you accept the [Atlassian Marketplace Vendor Agreement](#), the system submits your app for review by Atlassian's Developer Relations team.

## Importing workflows

### Custom fields in workflow imports

If your workflow contains custom fields that are disabled, the workflow importer will not create these fields unless they are enabled before importing. You will receive a warning about this. To fix this, you need to enable the missing custom fields before proceeding with the import.

1. Click on the highlighted **Custom Field Types & Searchers** plugin in the displayed warning. This opens the plugin in a new window and scrolls to the right place to make the necessary changes.
2. Click to expand the list of enabled modules.
3. Find the modules that are disabled and enable them.

After enabling the corresponding modules of the **Custom Field Types & Searchers** plugin, return to the summary page and proceed. You may need to refresh the page first. For information on installing apps, see [Viewing installed apps](#).

## Importing from Atlassian Marketplace

This procedure covers importing a workflow from Atlassian Marketplace.

1. Choose **Administration** (⚙️) > **Issues**.
2. Click on the **Workflows** section in the left-hand panel.
3. Select **Import > Import Workflow** in the top right of the screen.
4. The **From Atlassian Marketplace** option should be selected by default.
5. Find the workflow you want and click the **Select** button.
6. Follow steps 5 through 8 of the **Importing from a local instance** procedure.

## Importing from a local instance

This procedure covers importing a workflow from a local instance. ⓘ You must be logged in as System Administrator to perform this function.

1. Click on the **Workflows** section in the left-hand panel.
2. Select **Import > Import Workflow**.
3. Select a workflow from your computer to upload, and then click **Next**.
4. Jira automatically generates a workflow name, but you can change this if you like. Click **Next**.
5. Next, you are presented with a screen that details your workflow statuses, as shown below. You can map the steps of the workflow to your existing workflow statuses or create new statuses at this point. When you are finished, click **Next** to continue.
6. At the Preview of Import screen, click **Import** at the bottom of this screen to accept the changes and import the workflow.
7. Your workflow is imported and you are presented with a screen with additional configuration details. Click **Done** to exit this process.

ⓘ All custom fields will have brand new custom fields created. This is regardless of a custom field of the same name / type already existing. See:

 **JRASERVER-37358** - Workflow import creates duplicate custom fields **GATHERING INTEREST** for the request to improve this.

# Configuring workflow schemes

A workflow scheme defines a set of associations or mappings between a workflow and an issue type. Workflow schemes are associated with a project and make it possible to use a different workflow for every combination of project and issue type.

By default, projects use Jira's [system workflow](#). The default workflow scheme:

- Associates Jira's system workflow *Jira* with all issue types (available to the Jira project).
- Appears as the default workflow scheme for your selected project type.

In addition, you can share an existing project's workflow scheme when you are creating a new project by selecting **Create with shared configuration** in the [Project Creation Wizard](#). This allows you to reuse your existing schemes without having to recreate them for new projects. Keep in mind that changing shared workflow schemes will affect all projects that are using that theme.

This page describes how to configure workflows and issue type workflow associations in the scheme.

 To associate a workflow scheme with a project (part of activating a workflow), see [Managing your workflows](#).

For all of the following procedures, you must be logged in as a user with the [Jira Administrators global permission](#).

## Adding a workflow scheme

1. Choose **Administration**  > **Issues**. Select **Workflow Schemes** to open the Workflow Schemes page.
2. Click the **Add Workflow Scheme** button.
3. Enter the name and description of the new workflow scheme.
4. Click the **Add** button. The new workflow scheme is created.
5. Follow the instructions in [Configuring workflows for a workflow scheme](#) below.

## Configuring workflows for a workflow scheme

If your scheme is associated with a project, follow the instructions in [Configuring a workflow scheme associated with a project](#). Otherwise, follow the instructions in [Configuring a workflow scheme outside of a project](#).

### Configuring a workflow scheme associated with a project

 Jira's default workflow scheme cannot be modified. If you attempt to modify it, a copy of the scheme is created with the name of the project you are administering. You cannot configure a workflow scheme shared by multiple projects using this method; follow the instructions in [Configuring a workflow scheme outside of a project](#) instead.

1. Choose **Administration**  > **Projects**.
2. Select a project from the displayed list.
3. Click **Workflows** on the left of the **Project Summary** page. The **Workflows** page is displayed, indicating the current workflow scheme used by the project. Configure the workflow scheme using the table below.
4. At the **Publish Workflows** screen, click **Associate** to begin the migration process. Each issue has to be in a valid status. The valid statuses for an issue are defined by its workflow. This means that when changing a workflow, you may need to tell Jira the status for specific issues after the change.
5. A screen displays that indicates the progress of migrating all the project's issues to the updated scheme's workflows.

#### On this page:

- [Adding a workflow scheme](#)
- [Configuring workflows for a workflow scheme](#)
- [Editing, copying, and deleting workflow schemes](#)

6. Click **Acknowledge** to finish the process. A message displays letting you know that 'your workflows have been published.'

Configure the issue types for the workflow scheme as desired. This is not the same as editing the workflow (clicking the **Edit** button in the workflow diagram at the center of your screen). If you do that, you will be asked to publish your *draft workflow scheme*.

What do you want to do?	Instructions
Add a workflow to the scheme	<ol style="list-style-type: none"> <li>1. Click <b>Add Workflow</b>, and select <b>Import From Bundle</b> or <b>Add Existing</b>. After selecting Import From Bundle, you can select a workflow <b>From Atlassian Marketplace</b>. See <a href="#">Sharing your workflow</a> for more information.</li> <li>2. Select the desired workflow and issue types.</li> </ol>
Edit a workflow	Hover over the desired workflow and click the <b>Edit</b> button. See <a href="#">Working with workflows</a> for further instructions. The <b>Edit</b> button only displays if you have the edit permission.
Remove a workflow from the scheme	Click the cross icon under Operations to remove the workflow from the scheme.
Change the issue types associated with a workflow	<ol style="list-style-type: none"> <li>1. Click the <b>Assign</b> link under Issue Types for the desired workflow.</li> <li>2. Select the desired issue types in the dialog that appears.</li> <li>3. Click <b>Finish</b>.</li> </ol>
View the text-based representation of a workflow	Hover over the desired workflow, and click the <b>View as Text</b> link.
Change the workflow scheme associated with the project	Click the <b>Switch Scheme</b> button next to the scheme name. See <a href="#">Managing your workflows</a> for further instructions.

### Configuring a workflow scheme outside of a project

You can use this procedure to edit any workflow scheme in the system, including those shared by multiple projects. The workflow scheme can be either active or inactive.

- If your workflow scheme is associated with a project, you may want to follow the [instructions above](#) instead. When a workflow scheme is used by more than one project, you must use this configuration method.
  - When a workflow scheme is active, it creates a *draft workflow scheme* when you edit it.
1. Choose **Administration** () > **Issues**. Select **Workflow Schemes** to open the Workflow Schemes page.
  2. Click the **Edit** link under the Operations column for the desired workflow.
  3. Edit your workflow scheme, as described in the table below.
  4. If your workflow is active, you need to publish it to make your changes active.

What do you want to do?	Instructions
Add a workflow to the scheme	<ol style="list-style-type: none"> <li>1. Click <b>Add Workflow</b>, and select <b>Import From Bundle</b> or <b>Add Existing</b>. After selecting Import From Bundle, you can select a workflow <b>From Atlassian Marketplace</b>. See <a href="#">Sharing your workflow</a> for more information.</li> <li>2. Select the desired workflow and issue types.</li> </ol>

Remove a workflow from the scheme	Click the <b>Remove</b> link in the Operations column.
Change the issue types associated with a workflow	<ol style="list-style-type: none"> <li>1. Click the <b>Assign</b> link under Issue Types for the desired workflow.</li> <li>2. Select the desired issue types in the dialog that appears.</li> <li>3. Click <b>Finish</b>.</li> </ol>
View a representation of a workflow	Click either the text or diagram link next to the Workflow name.
Remove an issue type from the scheme	Click the <b>x</b> next to the name of the issue type to remove it.

### Editing, copying, and deleting workflow schemes

Choose **Administration**  > **Issues**. Select **Workflow Schemes** to open the Workflow Schemes page.

Operation	Instructions
Edit the name and description of a workflow scheme	Click the <b>Edit</b> link. Use inline edit mode click in the associated field to update the name and description.
Copy a workflow scheme	Click the <b>Copy</b> link to create a workflow scheme with the prefix "Copy of (name of current workflow)" and placed in the inactive workflow schemes.
Delete a workflow scheme	Click the <b>Delete</b> link and confirm the deletion. You cannot delete an active workflow scheme. You must first <a href="#">disassociate it</a> from all projects.

# Sharing your workflow

The new Workflow Sharing feature allows you to share your team's workflow with other teams in your organization on different Jira instances, or external parties in other organizations via the [Atlassian Marketplace](#). This feature allows you to easily share and use workflows that other people have published, or to move a workflow from staging to production in your own organization.

**Note:** For all of the following procedures, you must be logged in as a user with the **Jira Administrators** [global permission](#).

## On this page:

- [Exporting your workflow](#)
- [Uploading to Atlassian Marketplace](#)
- [Importing from Atlassian Marketplace](#)
- [Importing from a local instance](#)
- [Custom fields in workflow imports](#)

## Exporting your workflow

If you wish to share your Jira Workflow with another instance of Jira or upload it to the Atlassian Marketplace, you first need to download it. Follow this procedure.

1. Choose **Administration** (⚙️) > **Issues**.
2. Find the workflow you wish to share by clicking on the Workflows section in the left-hand panel.
3. Click **View** or **Edit** under the Operations column.
4. Select **Export > As Workflow**.
5. Click **Next** to continue.
6. In the Add Notes field, add any special configuration notes; for example, information about plugins that should be installed. Jira auto-populates these notes for you when it discards parts of your workflow (for example, plugins, post functions, conditions, validators).
7. Click **Export** and select a download location. Ensure the location is publicly accessible.

## Uploading to Atlassian Marketplace

To share your workflow with other Jira users, upload it to the Atlassian Marketplace.

1. Create an account on [Atlassian Marketplace](#).
2. Log in to the Atlassian Marketplace and choose **Manage apps**. See this page for more details: [Step-by-step Paid-via-Atlassian Listing](#).
3. Click **Create new app**.
4. Choose **My app is not directly installable**.
5. Ensure 'App Type' is listed as 'Not a Plugin.'
6. You will need to host the workflow on your own servers, and add information about where the workflow export can be accessed in the Binary URL textbox. This should be the location you specified in step 7 of the prior instruction set.
7. When you fill out the submission form, be sure to note the following:
  - a. The Summary field contains the information that will be displayed to users searching the Marketplace.
  - b. The Category for your workflow must be Workflow Bundles.
    - Choosing Workflow Bundles ensures other Jira users will have visibility to your workflow.
  - c. The App Key must be unique.
    - This is something that uniquely identifies your application; it will become the application URL.

You don't have to complete the form in one session. You can save your form and come back to it later. Once you accept the [Atlassian Marketplace Vendor Agreement](#), the system submits your app for review by Atlassian's Developer Relations team.

## Importing from Atlassian Marketplace

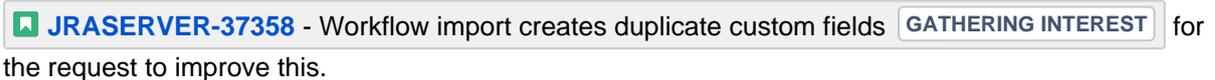
1. Choose **Administration** (⚙️) > **Issues**.
2. Click on the Workflows section in the left-hand panel.
3. Select **Import > Import Workflow** in the top right of the screen.
4. The **From Atlassian Marketplace** option should be selected by default.
5. Find the workflow you want and click the **Select** button.
6. Follow steps 5 through 8 of the 'Importing from a local instance' procedure.

## Importing from a local instance

This procedure covers importing a workflow from a local instance. For importing from Marketplace, see the procedure above, **Importing from Atlassian Marketplace**. You must be logged in as System Administrator to perform this function.

1. Click on the Workflows section in the left-hand panel.
2. Select **Import > Import Workflow**.
3. Select a workflow from your computer to upload, and then click **Next**.
4. Jira automatically generates a workflow name, but you can change this if you like. Click **Next**.
5. Next, you are presented with a screen that details your workflow statuses, as shown below. You can map the steps of the workflow to your existing workflow statuses or create new statuses at this point. When you are finished, click **Next** to continue.
6. You will be presented with a screen that presents a summary of the workflow changes, as shown below. Click **Import** at the bottom of this screen to accept these changes and import the workflow.
7. Your workflow is imported and you are presented with a screen with additional configuration details. Click **Done** to exit this process.

All custom fields will have brand new custom fields created. This is regardless of a custom field of the same name / type already existing. See:

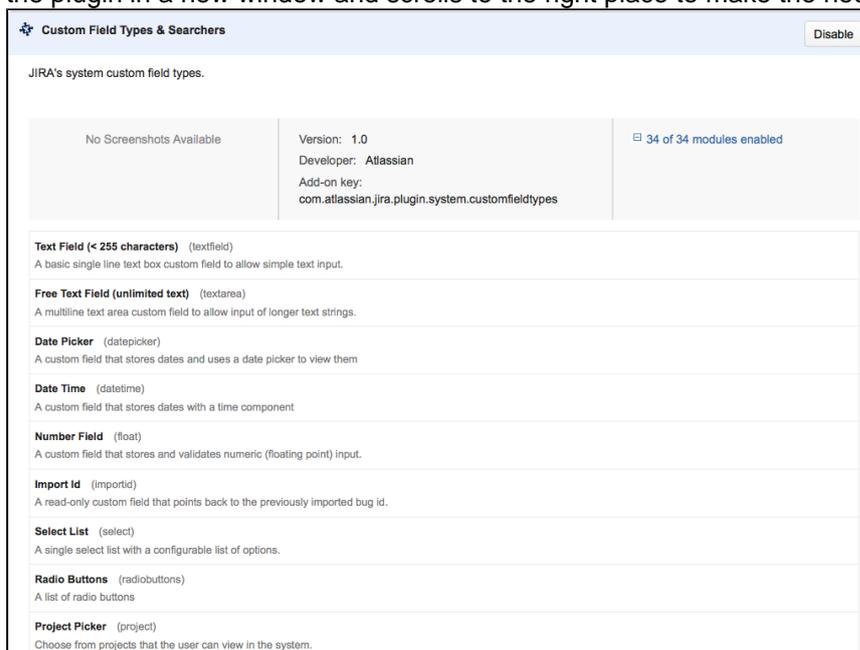
 **JRASERVER-37358** - Workflow import creates duplicate custom fields **GATHERING INTEREST** for the request to improve this.

## Custom fields in workflow imports

If the workflow that you are importing contains custom fields that are disabled, the workflow importer will not create these fields unless they are enabled before importing.

You will receive a warning about this. To fix this, you need to enable the missing custom fields before proceeding with the import.

1. Click on the highlighted Custom Field Types & Searchers plugin in the displayed warning. This opens the plugin in a new window and scrolls to the right place to make the necessary changes:



**Custom Field Types & Searchers** [Disable]

JIRA's system custom field types.

No Screenshots Available | Version: 1.0 | 34 of 34 modules enabled  
 Developer: Atlassian  
 Add-on key: com.atlassian.jira.plugin.system.customfieldtypes

- Text Field (< 255 characters)** (textfield)  
A basic single line text box custom field to allow simple text input.
- Free Text Field (unlimited text)** (textarea)  
A multiline text area custom field to allow input of longer text strings.
- Date Picker** (datepicker)  
A custom field that stores dates and uses a date picker to view them
- Date Time** (datetime)  
A custom field that stores dates with a time component
- Number Field** (float)  
A custom field that stores and validates numeric (floating point) input.
- Import Id** (importid)  
A read-only custom field that points back to the previously imported bug id.
- Select List** (select)  
A single select list with a configurable list of options.
- Radio Buttons** (radiobuttons)  
A list of radio buttons
- Project Picker** (project)  
Choose from projects that the user can view in the system.

2. Click to expand the list of enabled modules.

3. Find the modules that are disabled and enable them.

After enabling the corresponding modules of the Custom Field Types & Searchers plugin, return to the summary page and proceed. You may need to refresh the page first.

For information on installing apps, see [Viewing installed apps](#).

# Advanced workflow configuration

This page describes configuring transitions in Jira workflows. For information about the basics of workflows see [Working with workflow](#).

As a Jira administrator, you can control the following aspects of a transition's behavior:

- [Triggers](#) transition Jira issues when certain events occur in a connected development tool, such as Atlassian's [Bitbucket](#) or [Stash](#).
- [Conditions](#) check that a transition should be performed by the user.
- [Validators](#) check that any input to the transition (for example, by a user) is valid, *before* the transition is performed.
- [Post functions](#) carry out additional processing, *after* a transition is performed.
- [Properties](#) are key-value pairs that can be used to further customize transitions.

Also on this page:

- [Customize how transitions appear](#)
- [Global transitions](#)

## Triggers

Jira administrators can configure triggers in Jira workflows that respond to events in your linked development tools. This allows you to set up your development tools and Jira workflows so that, for example, when a developer creates a branch to start work on an issue in Atlassian's [Bitbucket](#) or [Stash](#), the issue will automatically be transitioned from 'Open' to 'In progress'.

📘 If you haven't set up a trigger before or you want to learn about triggers in more detail, see our guide on triggers here: [Configuring workflow triggers](#). The guide also shows you how to configure a workflow with triggers, similar to this sample development workflow: [Development Workflow with Triggers](#) (from Atlassian Marketplace).

## Configure triggers

To see, or to set, triggers for a transition, edit the workflow that contains the transition, select the transition, then click **Triggers** in the properties panel for the transition.

### To add a trigger to a transition:

1. Log in as a user with the 'Jira Administrators' [global permission](#).
2. Choose **Administration** (⚙️) > **Issues**. Select **Workflows** to open the Workflows page, which displays all of the workflows in your system.
3. Click **Edit** for the workflow that has the transition you wish to change.
4. In the Workflow Designer, select the transition.
5. Click **Triggers** in the properties panel to show the triggers configured for the transition.
6. Click **Add trigger** on the **Trigger** tab to configure a trigger.

## Conditions

Conditions control whether a transition should be executed by the user. As examples, conditions can be used to:

- allow only the reporter to execute a transition.
- allow only users with a certain permission to execute a transition.
- allow execution only if code has, or has not, been committed against this issue.

If a condition fails, the user will not see the transition button on the 'View issue' page, and so will not be able to execute the transition.

Conditions cannot validate input parameters gathered from the user on the transition's screen you need a [validator](#) to do this.

The following sections describe:

- [Adding a condition](#)
- [Grouping conditions](#)

## Adding a condition

To add a condition to a transition, edit the workflow that contains the transition, select the transition, then click **Conditions** in the properties panel for the transition.

### To add a condition to a transition:

1. Log in as a user with the 'Jira Administrators'[global permission](#).
2. Choose **Administration**  > **Issues**. Select **Workflows** to open the Workflows page, which displays all of the workflows in your system.
3. Click **Edit** for the workflow that has the transition you wish to change.
4. In the Workflow Designer, select the transition:
5. Click **Conditions** in the properties panel.

On the **Condition** tab, you can see any conditions that have already been set.

When you click **Add condition**, you can choose from the available conditions, and set any necessary parameters for the condition. Additional conditions may be available from installed apps. or you can create your own conditions using the [app system](#); see the [Workflow app Modules](#) for details.

Note that you can also edit the transition in ['text' mode](#).

## Grouping conditions

You can construct complex conditions by grouping and nesting conditions. Change any condition into a group by clicking the 'Add grouped condition' icon for the condition. Now you can add further conditions to this new group, as described [above](#).

You can toggle the logic for how the conditions in a group are applied between **All** and **Any**.

## Validators

Validators check that any input made to the transition is *valid before* the transition is performed. Input can include that gathered from the user on the transition's screen.

If a validator fails, the issue does not progress to the destination status of the transition, and the transition's [post functions](#) are not executed.

## Adding a validator

To add a validator to a transition, edit the workflow that contains the transition, select the transition, then click **Validators** in the properties panel for the transition.

### To add a validator to a transition:

1. Log in as a user with the 'Jira Administrators'[global permission](#).
2. Choose **Administration**  > **Issues**. Select **Workflows** to open the Workflows page, which displays all of the workflows in your system.
3. Click **Edit** for the workflow that has the transition you wish to change.
4. In the Workflow Designer, select the transition:
5. Click **Validators** in the properties panel.

On the **Validator** tab, you can see any validators that have already been set.

When you click **Add validator** you can choose from the available validators and set any necessary parameters for the validator.

Note that you can also edit the transition in ['text' mode](#).

## Post functions

Post functions carry out any additional processing required after a transition is executed, such as:

- updating an issue's fields
- generating change history for an issue
- adding a comment to an issue
- generating an event to trigger email notifications

The following sections describe:

- [Essential post functions](#)
- [Optional post functions](#)
- [Using post functions with the initial transition](#)
- [Using a post function to set a field](#)
- [Using a post function to send HipChat notifications](#)
- [Using a post function to send email notifications](#)

## Essential post functions

Every Jira transition has the following essential post functions, which are performed in this order:

1. Set issue status to the linked status of the destination workflow status.
2. Add a comment to an issue if one is entered during a transition.
3. Update change history for an issue and store the issue in the database.
4. Reindex an issue to keep indices in sync with the database.
5. Fire an event that can be processed by the listeners.

These essential post functions cannot be deleted from a transition or reordered. However, you can insert other (optional) post functions between them.

## Optional post functions

Jira includes several optional post functions that can be added to transitions.

Optional post function	Description
Assign to Current User	Assigns the issue to the user who is executing the transition.   This post function is ignored unless the user has the <a href="#">Assignable User permission</a> . Create a condition to give the logged-in user this permission before executing the transition.
Assign to Lead Developer	Assigns the issue to the component lead, if one exists, or project lead.
Assign to Reporter	Assigns the issue to the user who created the issue.
Create Perforce Job Function	Creates a Perforce Job (if required) after completing the workflow transition.

Notify HipChat	Sends a notification to one or more HipChat rooms. See <a href="#">Using a post function to send HipChat notifications</a> for more information.
Trigger a Webhook	Triggers the specified webhook after completing the workflow transition.  When you add this post function, you will be asked to specify a webhook. This webhook must already be defined in Jira (see <a href="#">Managing webhooks</a> ).
Update Issue Field	Updates one of the issue's fields to a given value. Fields that can be updated include: <ul style="list-style-type: none"> <li>• Assignee</li> <li>• Description</li> <li>• Environment</li> <li>• Priority</li> <li>• Resolution</li> <li>• Summary</li> <li>• Original Estimate</li> <li>• Remaining Estimate</li> </ul> <p><b>i</b> This post function cannot update <a href="#">custom fields</a> and must be positioned after the other optional post functions.</p>

Additional post functions may be available from installed apps. or you can create your own post functions using the [app system](#); see the [Workflow app Modules](#) for details.

#### Adding a post function

To add a post function to a transition, edit the workflow that contains the transition, select the transition, then click **Post functions** in the properties panel for the transition.

##### To add a post function to a transition:

1. Log in as a user with the 'Jira Administrators'[global permission](#).
2. Choose **Administration**  > **Issues**. Select **Workflows** to open the Workflows page, which displays all of the workflows in your system.
3. Click **Edit** for the workflow that has the transition you wish to change.
4. In the Workflow Designer, select the transition:
5. Click **Post functions** in the properties panel.

On the **Post functions** tab, you can see any post functions that have already been set. When you click **Add post function** you can choose from the available post functions, and set any necessary parameters. Options for editing or deleting a post function, and for changing the execution order, are at the right of the tab (hover there to see them).

Note that you can also edit the transition in ['text' mode](#).

#### Using post functions with the initial transition

You can add post functions to a workflow's initial transition when you need to perform processing tasks such as setting a particular field's value when an issue is created. The initial transition is called 'Create' (if you created a blank workflow) or 'Create Issue' (if you copied the system workflow).

Jira includes the following [essential post functions](#) that are specific to a workflow's initial transition and that are performed in this order:

1. Create the issue.
2. Fire an event that can be processed by the listeners.

The following optional post functions are available specifically for the initial transition:

Optional post function (initial transition only)	Description
Create Comment	Adds a comment to an issue if one is entered during a transition.
Update Issue Status	Sets the issue's status to the linked status of the destination workflow status.
Store Issue	Stores updates to an issue (no change history is created).

Additionally, the standard [optional post functions](#) can also be added to an initial transition,

Optional post functions added to the Create transition must be placed *before* the 'Create the issue originally' post function.

If you wish, you can configure the initial status for your workflow to go to a different initial transition. See [Configuring the initial status](#) for details.

#### Notes

If you need to set the 'Resolution' field when creating an issue, add the 'Update Issue Field' post function *after* the 'Create the issue' post function and *after that*, use the 'Store Issue' post function. The 'Store Issue' post function is useful for setting the Resolution field during issue creation.

However, only use the Store Issue post function where necessary, since it:

- does not generate change history
- is unable to persist fields that have a one-to-many relationship with the issue (for example, 'Version' or 'Component')

### Using a post function to set a field

You can use the 'Update Issue Field' post function to set the value of an issue's field after a particular transition is executed.

For example, you might want a transition that moves the issue to a *closed* status to automatically set the 'Resolution' field.

#### Example: Using a post function to set the Resolution field:

1. Edit the workflow that has the transition, and drag from status to another to create a new transition.
2. Select either **None** or a screen that does not contain the **Resolution** field:

3. [Add a new post function](#) of type 'Update Issue Field' and:
  - a. Select **Resolution** from the **Issue Field** list.
  - b. Select a suitable resolution from the **Field Value** list.

To create a transition that clears the **Resolution** field, follow the same steps above for adding an 'Update Issue Field' post function to your transition. However, select **None** from the **Field Value** list.

The list of post functions for this transition includes the following statement:

- The **Resolution** of the issue will be **cleared**.

Each time one of these transitions is executed, the **Resolution** of the issue is automatically set or cleared, as specified in these post functions.

### Using a post function to send HipChat notifications

You can use a 'Notify HipChat' post function to send a notification to one or more HipChat rooms whenever an issue passes through a transition with this post function. You can also add a JQL query to the 'Notify Hipchat' post function to filter for the issues that will trigger the HipChat notification.

#### To send HipChat notifications:

1. Create or edit your transition.
2. [Add a new post function](#) of type 'Notify HipChat'.
3. On the 'Add Parameters to Function' page:
  - a. Optionally, specify a JQL query. Only issues that match the query will send notifications. Leave this field empty to send notifications to *all* issues that pass through this transition.
  - b. Select the HipChat rooms you want to link with your workflow transition.

### Using a post function to send email notifications

Use the 'Fire an event that can be processed by the listeners' post function to fire the 'Generic Event', which is a built-in [Jira event](#) that can be used to trigger the sending of [email notifications](#) after a particular transition is executed.

Alternatively, you could fire a [custom event](#) that you've created specifically for this transition.

When a transition is performed, Jira will:

- Look up the [notification scheme](#) associated with the issue's project and identify the users associated with the fired event;
- Send an email notification to each user.

 The fired event is also propagated to all registered [listeners](#).

#### Example: Using a post function to fire the Generic Event to send email notifications:

1. Create or edit your transition.
2. Click the transition's **Post Functions** tab and edit the 'Fire an event that can be processed by the listeners' post function.
3. Select **Generic Event** from the list of events.

## Transition properties

Properties are key-value pairs that can be used to further customize transitions. For example, transition properties can help to extend a copiedsystem workflow to allow language translations.

#### To view and edit the properties of a transition:

1. Select a transition in the diagram.
2. Click **Properties** in the Properties panel.
3. Either:
  - Add a new property to the transition.
  - Delete a property, by clicking the icon to the right of the property.

**i Important**

It is not possible to edit a transition's properties on this page. To change any property's key or value (or both), you must first delete the property you wish to change and add the new updated property.

Note that you can also edit the transition in ['text' mode](#).

It is possible to implement restrictions on transitions using transition properties. For more information, see [Workflow properties](#).

## Customize how transitions appear

When viewing an issue, most of the operations and workflow transitions are available from a row of buttons at the top of the issue.

### To change the number of transition buttons from the default of two:

By default, the first two transitions appear as separate buttons in the set of transition buttons. Additional transitions appear in the **Workflow** menu. The order in which these buttons appear is based on the order defined in the system workflow.

1. Shutdown Jira.
2. Edit the `jira-config.properties` file in your [Jira application home directory](#). See [Making changes to the jira-config.properties file](#) for more information.
3. Change the value of 'X' in the `ops.bar.group.size.opsbar-transitions = X` property of this file to be the number of transition buttons required *before* the **Workflow** menu.
  - If this property does not exist in your `jira-config.properties` file, add it. Otherwise, a default value of 2 is assumed.
4. Save the updated `jira-config.properties` file.
5. Restart Jira.

### To change the order of transition buttons:

To change the order of transition buttons, including additional transitions in the **Workflow** menu, add the property key `opsbar-sequence` to each [workflow transition](#) that you wish to reorder. Each `opsbar-sequence` property key requires a property value that defines the order of the transition action on issue views.

1. Go to the transition's properties, as described in [Transition properties](#) above.
2. Type `opsbar-sequence` into the **Property Key** field, under 'Add New Property'.
3. Type a value in the **Property Value** field. The value must be a positive integer (starting at '0'); it defines the order of the transition buttons on issue views.
  - Consider using a sequence of `opsbar-sequence` property values like 10, 20, 30... to allow new transitions to be easily added later.
4. Click **Add**.

• Adding the `opsbar-sequence` property to a workflow transition does not change the order of these transitions in the workflow in **Text** edit mode. The addition of this property only affects the order of transitions on the **View issue** page.

## Global transitions

Global transitions allow any status in a workflow to transition to a particular status.

You can add a global transition:

- When creating a new status (adding an existing status) check the **Add global transition to status** option.
- By selecting a status and checking **Allow all statuses to transition to this one** in the properties panel for the status.



To create two global transitions that point to the same destination step:

1. From the workflow designer, create the first global transition as normal by selecting a step and checking "Allow all statuses to transition to this one"
2. Create the second global transition on any *other* step that does not currently have a global transition pointing to it
3. Then from text editor, select the second global transition that you created
4. Click on the 'Edit' button and change the 'Destination Step' to the same step that you selected for your first global transition, and then click 'Update'

# Working in text mode

Text mode is an advanced way of working with workflows, and it shows the difference between steps and statuses. In text mode, you work directly with steps.

For all of the following procedures, you must be logged in as a user with the **Jira Administrators** [global permission](#) and start from the Workflows page.

## On this page:

- [Basic procedures](#)
- [Advanced procedures](#)

## To open a workflow in text mode

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Workflow** to open the Workflows page, which displays all of the workflows in your system.
3. Select **Edit** to open the workflow for editing.
4. Select the **Text** button to edit in text mode. A list of existing steps that comprise the workflow and each step's **Linked Status** and **Outgoing Transitions** (under **Transitions (id)**), is shown.
5. Follow the relevant procedure below to edit the workflow.

**i** Note that some workflow elements cannot be edited in text mode, such as global transitions. If you cannot change a workflow element, try editing in Diagram mode.

## Basic procedures

### Editing a step

Click the following link of any step:

- **Add Transition** to add an **Outgoing Transition** to that step.
- **Delete Transitions** to delete one or more **Outgoing Transitions** of that step.
- **Edit** to edit the step's **Step Name** or **Linked Status**.
- **View Properties** to view and edit the step's **Properties**.
- **Delete Step** only available if the step has no **Incoming Transitions**.

### Adding a step

The **Add New Step** form appears below the list of steps when you are editing an inactive workflow.

#### To add a new step to a workflow:

1. In the **Step Name** field, type a short name for the step.
2. In the **Linked Status** field, select the status that corresponds to this step.  
**i** Each status can only correspond to one step in each workflow.
3. Click the **Add** button. Your new step appears in your workflow's list of steps in **Text** edit mode.

**i** If you do not see **Add New Step**, this means that all available statuses defined in your Jira installation have been used in your workflow and you need to [define a new status](#).

### Deleting a step

*A step can only be deleted if it has no incoming transitions.*

Click the **Delete Step** link that corresponds to the relevant step.

**i** This link is not displayed if the step has no incoming transitions or if it only has incoming **Global Transitions**.

### Adding a transition

1. Identify the step from which your new transition will originate, and click the **Add Transition** link next to the step.
2. In the Transition Name field, type a short name for the transition.
  - This name will be shown to users on the relevant transition button on the View issue page.
3. (Optional) In the Description field, type a short description of the purpose of the transition.
4. In the Destination Step field, choose the step to which issues will move when this transition is executed.
5. In the Transition View field, select either:
  - No view for transition choose this if you do not need to prompt the user for input before the transition is executed (i.e. the transition will occur instantly when the user clicks the transition).
  - The name of a [screen](#) that will be shown to users, asking for input before the transition is executed. You can choose one of Jira's default screens or any other screen you have created. If no existing screen is suitable, you may wish to create a new screen for the transition.

### Editing or deleting a transition

1. In the Transitions (id) column, click the link of the **Outgoing Transition** of the step you wish to edit. The Transition page is displayed.
2. From this point, you can:
  - Click the buttons at the top of the page to edit or delete the transition.
    - Note: You will only be able to delete a transition if this step has at least one outgoing transition indicated in the Workflow Browser section. In the image above, this is not the case.
  - Click **View Properties** to edit the transition's properties. See [Advanced workflow configuration](#) for details.
  - Add a new condition, validator, or post function. See [Advanced workflow configuration](#) for details.

### Advanced procedures

#### Preventing issues from being edited

You can use a workflow step's properties to prevent issues from being edited in a particular workflow step. For example, in a copied system workflow, **Closed** issues cannot be edited, even by users who have the Edit Issue [project permission](#).

#### Note:

- Issues that cannot be edited cannot be updated using bulk edit.
- You can only edit the properties of a workflow's step if that workflow is editable (i.e. if that workflow is either [inactive](#) or a draft of an active workflow).

#### To stop issues from being editable in a particular workflow step or to set any property of a step:

1. Click the **View Properties** link that corresponds to the relevant step.
2. In the **Property Key** field, type: `jira.issue.editable` (or any other **Property Key** you wish to add).
3. In the **Property Value** field, type: `false` (or any other **Property Value** you wish to add).
4. Click the **Add** button.

#### Note:

- It is not possible to edit a step's properties on this page. To change any property's key or value, you must first delete the property you wish to change and then add the new, updated property.
- It is possible to implement restrictions on steps using step properties. For more information, see [Workflow properties](#).

#### Using a screen with a transition

When a user clicks a particular transition, a screen can be used to gather input from the user before the transition is executed.

#### Example: using a screen to set the Resolution field

For a particular step in a workflow, you might need to create a transition that moves the issue to a Closed status. To do this:

1. Create or edit your transition.
2. Select the **Resolve Issue Screen** in the **Transition View** field.
3. Click **Add** when you are finished editing the workflow transition. You will be back on the **Text** view screen of the project's workflow.

**See also:**

- [Working with workflows](#)
- [Advanced workflow configuration](#)

# Adding a custom event

Jira uses an event-listener mechanism to alert the system that something has happened, and to perform appropriate action (e.g. send an email notification) based on the event that has occurred. Every *issue operation* within Jira is associated with a particular *event*- e.g. the `Issue Created` event is fired when an issue has been created. A *listener* can execute a specified action once it has been notified that a particular event has been fired. For example, the `MailListener` can send an `Issue Created` email to a list of recipients defined in the appropriate *notification scheme*, whenever an issue is created.

- [System events](#)
- [Custom events](#)
- [Configuring notifications for a custom event](#)

Some events are fired by Jira internally e.g. an `Issue Updated` or `Issue Moved` event. Other events are fired from within *workflow transition post functions* e.g. an `Issue Resolved` event, or a custom event (see below).

There are two types of events within Jira:

- **System** System events are used throughout Jira internally, and cannot be added or deleted. You can, however, make them Inactive (see below).
- **Custom** Custom events are used to generate an email notification (or invoke a listener) from a particular workflow transition's post function. You can add and delete as many custom events as you need. Note that only *inactive* custom events can be deleted.

An event can be in either of the following states:

- **Active** the event is associated with at least one notification scheme or workflow transition post function.
- **Inactive** the event is not associated with any notification schemes or workflow transition post functions.

Note that the event state does not indicate whether the event is able to be fired. A custom event will only be fired if it is associated with a transition post function for an active workflow (see [Managing your workflows](#)).

## System events

Jira's built-in system events are:

Issue created	An issue has been entered into the system.
Issue updated	An issue has had its details changed.
Issue assigned	An issue has been assigned to a new user.
Issue resolved	An issue has been resolved (usually after being worked on and fixed).
Issue closed	An issue has been closed. (Note that an issue may be closed without being resolved; see <a href="#">statuses</a> ).
Issue commented	An issue has had a comment added to it.

Issue comment edited	An issue's comment has been modified.
Issue reopened	An issue has been re-opened.
Issue deleted	An issue has been deleted.
Issue moved	An issue has been moved into this project.
Work logged on issue	An issue has had hours logged against it (i.e. a worklog has been added).
Work started on issue	The Assignee has started working on an issue.
Work stopped on issue	The Assignee has stopped working on an issue.
Issue worklog updated	An entry in an issue's worklog has been modified.
Issue worklog deleted	An entry in an issue's worklog has been deleted.
Generic event	The exact nature of this event depends on the workflow transition post function(s) which invoke it. As with custom events, you can use the generic event to generate an email notification (or invoke a listener) from a particular workflow transition's post function (see <a href="#">Working with workflows</a> ).

## Custom events

You can fire a custom event from a custom transition post function in a custom workflow. The appropriate listeners will be alerted of the custom transition by the firing of this event. For example, the associated notification scheme can be configured to notify users of the workflow transition based on the firing of this custom event.

## Configuring notifications for a custom event

Custom events are most commonly used to generate notifications for custom workflow transitions. For example, your organization might need you to modify the [default workflow](#) by adding a workflow step called 'QA\_Inspection' (e.g. between **Resolve Issue** and **Close Issue**). You would typically also need to generate an email notification to the QA team whenever an issue progresses to the 'QA\_Inspection' step of the workflow.

There are three overall steps to achieve this:

1. Add a custom event to the system (e.g. 'Issue Awaiting QA').

2. Configure the notification scheme to send an email when the custom event is fired.
3. Configure the workflow transition post function to fire the custom event.

### Adding a custom event

1. Log in as a user with the **Jira Administrators** [global permission](#).
2. Choose **Administration**  > **System**. Select **Advanced > Events** to open the View Events page.
3. In the Add New Event form at the bottom of the page, add a name and description for the custom event.
4. In the Template field, select the default email template to be associated with the event.
5. Click the **Add** button.

The custom event must be associated with a default email notification template. A notification scheme configured to notify users of this event will use this email template when sending the notification.

The custom event will appear in the list of events defined within the system. Initially, the event will be marked as inactive, as it is not associated with a notification scheme or workflow post function.

### Configuring the notification scheme to send mail

1. Log in as a user with the **Jira Administrators** [global permission](#).
2. Choose **Administration**  > **System**. Select **Advanced > Events** to open the View Events page.
3. Select the notification scheme to edit, by clicking the notification scheme's name or its **Notifications** link (under Operations).
4. Add the recipients for the custom event as required. See [Creating a notification scheme](#) for more information.

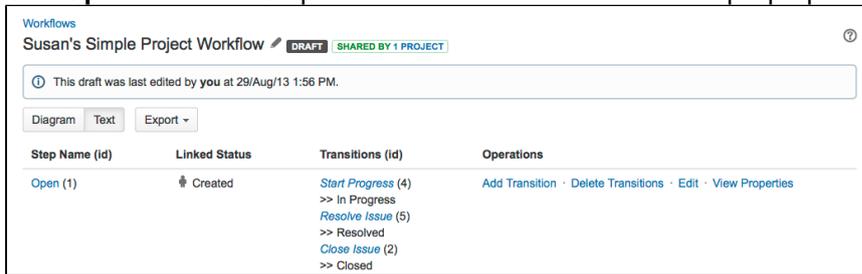
### Configuring a post function to fire the custom event

1. Log in as a user with the **Jira Administrators** [global permission](#).
2. Choose **Administration**  > **Issues**. Select **Workflows** to open the Workflows page, which displays all of the workflows in your system.
3. Navigate to workflow transition post function screen to be edited. See [Working with workflows](#) and [Advanced workflow configuration](#) for more information.
4. Update the post function to fire the custom event.
5. Activate or associate the workflow (and scheme) with the appropriate project. See [Managing your workflows](#) for more information.

# Configuring the initial status

Use this procedure to configure the initial status for your workflow. You can start off with an active workflow, which you can then switch to draft mode, or any other workflow in your system.

1. Click **Open** under the Step Name column to view or edit a step's properties:

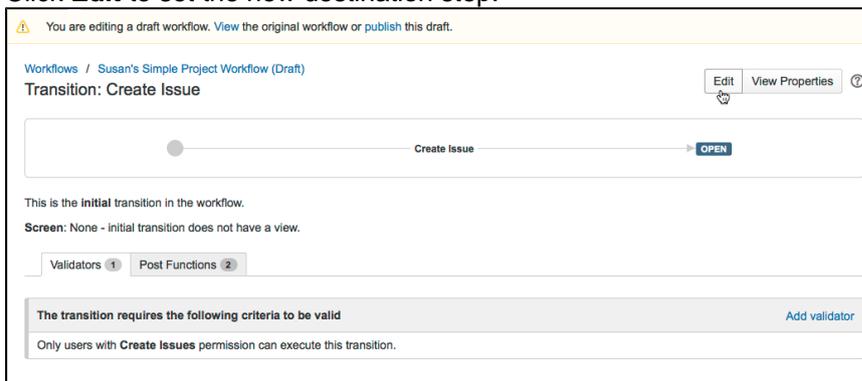


2. Click the **Create Issue** incoming transition:

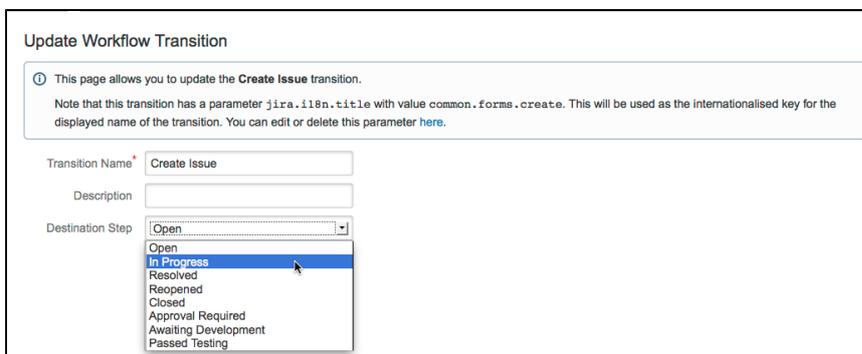


**Note:** If you happen to be in an active workflow, which you cannot edit, you will be asked to switch to a draft workflow to continue.

3. Click **Edit** to set the new destination step:



4. Select a new **Destination Step**, and then click **Update** to save it:



5. When a new issue is created, it will go straight to the **In Progress** step.

# Configuring workflow triggers

 You must have Jira Software to start using workflow triggers.

Triggers are a powerful tool for keeping your Jira issues synchronized with the information in your development tools (Fisheye/Crucible, Bitbucket and GitHub). Instead of relying upon developers to manually update the status of issues after committing code, completing reviews, creating branches, etc, you can configure triggers in your workflow to automatically transition issues when these events occur in your development tools. For example, you could configure a trigger to automatically transition an issue from 'To Do' to 'In Progress' when a branch is created.

This page will help you get started using triggers. We will show you how to set up triggers in a workflow and demonstrate how an automatic transition works. We will also provide some guidelines on how to best configure a trigger and help you troubleshoot your triggers.

## On this page:

- [Before you begin](#)
- [Guide: Setting up triggers](#)
- [Understanding triggers](#)
- [Troubleshooting](#)

## Before you begin

Before you can start using triggers, you need to connect your development tools to Jira Software. At a minimum, you will need a Jira Server instance, plus at least one of the following:

- Bitbucket Server ([all current versions](#))
- Fisheye/Crucible ([all current versions](#))
- GitHub Enterprise 11.10.290 (or later)
- Bitbucket
- GitHub

For instructions on how to connect these tools to Jira, see [Integrating with development tools](#). This page also includes details on other functionality you can enable by connecting the various development tools Atlassian offer.

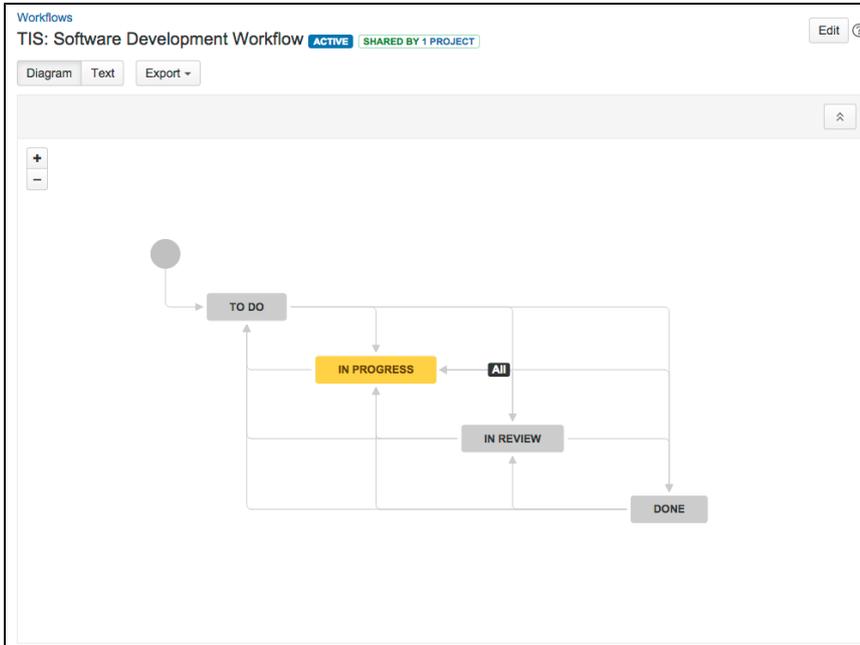
## Guide: Setting up triggers

In this example, you will be configuring a Jira workflow with triggers. By the end of this section, you will have an understanding of how to configure triggers and what a typical development workflow with triggers looks like.

- [Introduction](#)
- [Step 1. Create/Edit a workflow](#)
- [Step 2. Add a trigger to a transition](#)
- [Step 3. Test the trigger](#)
- [Step 4. Add the rest of the triggers](#)

### Introduction

The screenshot and table below show a workflow and triggers similar to what you will be configuring. They reflect the typical interactions between Jira and development tools in a software development lifecycle. Jira Software, Bitbucket Server and Fisheye/Crucible (3.5.2) are used for this example, but you can configure something similar using any of the supported development tools.



Transition	Triggers
Start progress (To Do In Progress)	Branch created (Bitbucket Server) Commit created (Bitbucket Server)
Start review (In Progress In Review)	Pull request created (Bitbucket Server) Pull request reopened ((Bitbucket Server)  Review started (Crucible)
Restart progress (In Review In Progress)	Pull request declined (Bitbucket Server) Review rejected (Crucible) Review abandoned (Crucible)
Done (In Review Done)	Pull request merged (Bitbucket Server) Review closed (Crucible)

**Step 1. Create/Edit a workflow**

The easiest way to create a software development workflow is to [create a new project](#), choosing a relevant project type. This will set up your new project with this software development workflow, which is identical to the one shown above.

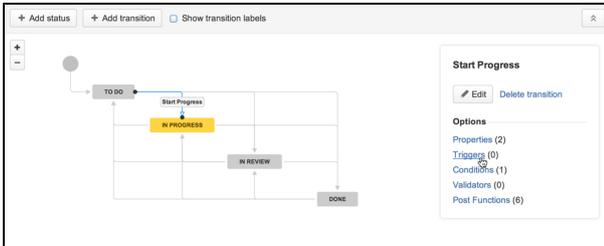
If you already have a similar workflow, navigate to it and edit it: Jira administration console > **Issues** > **Workflows** > **Edit**

**Step 2. Add a trigger to a transition**

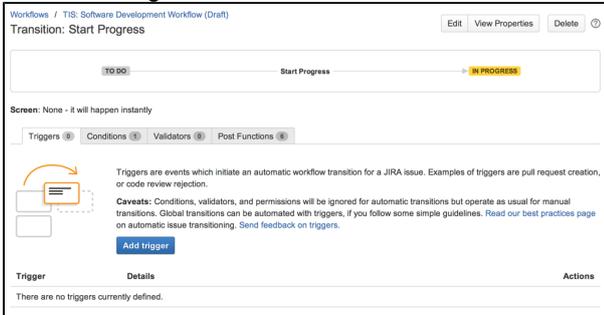
We'll start by adding a 'Commit created' trigger to the 'Start progress' transition. Ensure that you are editing (not viewing) the workflow.

1. Select the **Start progress** transition in the workflow, i.e. the line from 'To Do' to 'In Progress'. A panel will display on the right, showing the details of the transition.

*Related topic: [Why you shouldn't configure triggers on global transitions](#)*



2. Click **Triggers** in the panel. The 'Transition: Start Progress' screen will display with the 'Triggers' tab showing.

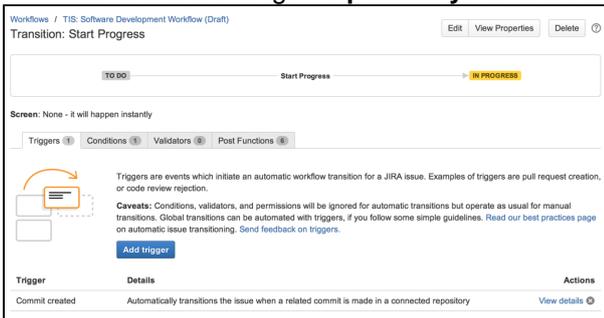


3. Click **Add trigger**, then select **Commit created** in the dialog that appears. A diagnostics window will display you'll notice that the trigger will be added for all development tools that Jira is connected to.

*Related topic: [How to enable different events for triggers](#)*

4. Click **Add trigger** to add the trigger. It will appear in a list at the bottom of the 'Triggers' tab. You can check whether it is working by clicking **View Details**.

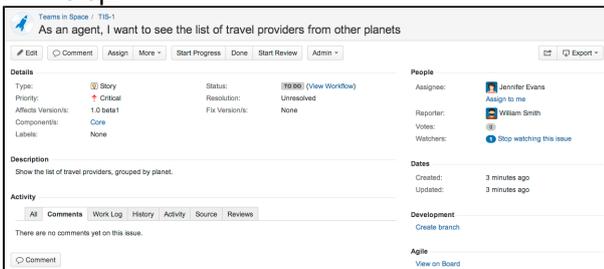
That's it! Don't forget to **publish your draft workflow**.



### Step 3. Test the trigger

Now that you have added the 'Commit created' trigger to the 'Start progress' transition, let's test it by making a commit.

1. Create an issue in your Jira project. This project needs to be using the workflow that you just edited. The status of your new issue should be 'To Do'. Take note of the issue key, as you'll need it for the next step.



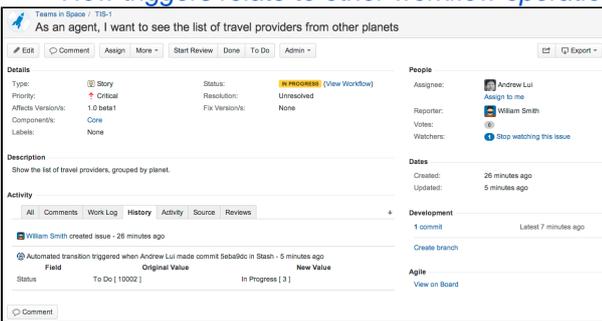
2. Commit some code to your Bitbucket repository. You can commit anything, however you will need to include the issue key in your commit message. In this example, the issue key is TIS-1, which is referenced in the commit message shown in the screenshot.

*Related topic: [Referencing a Jira issue in a commit, branch, pull request, or review](#)*

```
alui:tis-core alui$ git commit -am "TIS-1 Initial commit"
[master (root-commit) 274323b] TIS-1 Initial commit
 1 file changed, 7 insertions(+)
 create mode 100644 provider-list.html
alui:tis-core alui$ git push origin master
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 270 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To http://admin@localhost:7990/scm/tis/tis-core.git
 * [new branch]      master -> master
alui:tis-core alui$
```

3. Check your issue in Jira again. The status should have changed from 'To Do' to 'In Progress'. If you click the **History** tab or **Activity** tab, you can see the automatic transition that changed the issue's status.

*Related topics: [How the user is mapped from the development tool to Jira](#);  
[Event handling and event limits](#);  
[How triggers relate to other workflow operations/constraints](#)*



**Step 4. Add the rest of the triggers**

Now that you've added and tested a trigger, follow the same process to add the rest of the triggers in the [list above](#).

Don't want to set all of this up? Good news! You can download a similar workflow (pre-configured with triggers) from the Atlassian Marketplace: **download 'Development Workflow with Triggers'**.

**✔ Congratulations! You have now set up a workflow with triggers.**

- If you are having problems configuring your trigger or getting it working, check the [Troubleshooting section](#) below.
- If you want to learn more about how triggers work, see the [Understanding triggers section](#) below.

**Understanding triggers**

The following topics explain how triggers work in more detail, so you can use them more effectively.

**Trigger events**

Events (e.g. Commit created) are made available for triggers by integrating Jira with particular development tools. The table below lists the events that are enabled for each development tool.

Dev tool	Bitbucket, GitHub, GitHub Enterprise	Crucible	Fisheye
----------	--------------------------------------	----------	---------

Events			
	<ul style="list-style-type: none"> <li>• Pull request created</li> <li>• Pull request merged</li> <li>• Pull request declined (Bitbucket only)</li> <li>• Pull request reopened (Bitbucket Server only)</li> <li>• Commit created</li> <li>• Branch created</li> </ul>	<ul style="list-style-type: none"> <li>• Review started</li> <li>• Submitted for approval</li> <li>• Review rejected</li> <li>• Review abandoned</li> <li>• Review closed</li> <li>• Review summarized</li> </ul>	<ul style="list-style-type: none"> <li>• Commit created</li> <li>• Branch created</li> </ul>

 There is a known issue where the 'Branch created' event isn't supported for GitHub, which is being tracked under

 [JSWSERVER-14473](#) - Implement 'Create Branch' feature in DVCS connector plugin for Github integration CLOSED

please keep this in mind when configuring trigger events.

### Triggers and global transitions

We recommend that you *do not configure triggers for global transitions*, unless you are confident that you understand exactly how the trigger will affect the behavior of the issue.

A global transition allows any status in a workflow to transition to a particular status. This is represented in the workflow viewer/editor by a black **All** lozenge pointing to the status that the global transition targets. For more information about global transitions, see [Advanced workflow configuration](#).

Configuring triggers for global transitions can often result in an issue unexpectedly transitioning to the target status for the global transition. For example, consider if you configured a 'Commit created' trigger for the global transition to the 'In Progress' status. Committing code can happen at many stages during an issue's lifecycle (e.g. writing the initial code, changing code after a review, etc). This could result in the issue incorrectly transitioning to 'In Progress' out of a number of statuses, like 'In Review' or 'Done'.

*Tip: If you do use global transitions in your workflow, you will probably have multiple transitions into a status. This means that users will have multiple workflow options on an issue (e.g. both 'Start Progress' and 'In Progress'). To hide options, add the 'Hide transition from user' condition to the relevant transitions.*

### Referencing a Jira issue in a commit, branch, pull request, or review

The table below describes how to reference a Jira issue in a commit, branch, pull request, or review, so that these events will trigger transitions for the issue (provided that you have set up triggers on the transitions).

Event	Instructions
<b>Create commit</b>	<p>Include the issue key in the commit message.</p> <p>For example, a commit message like this "TIS-1 Initial commit" will automatically transition the TIS-1 issue from 'To Do' to 'In Progress'.</p>
<b>Create branch</b>	<p>Include the issue key in the branch name, when you create the branch.</p> <p>For example, if you name your branch "TIS-2-feature", it will automatically transition the TIS-2 issue from 'To Do' to 'In Progress'.</p>
<b>Create /Reopen /Decline Merge pull request</b>	<p>Ensure that the pull request includes commits that reference the issue (in their commit messages).</p> <p>For example, if you create a pull request that has "TIS-3" in the title, it will automatically transition the "TIS-3" issue from 'In Progress' to 'In Review'. If you reopen, decline or merge the pull request, it will also transition the "TIS-3" issue accordingly.</p>

<b>Start/Reject /Abandon /Close review</b>	<p>Include the issue key in the review title, when you create the review.</p> <p>For example, if you name your review "TIS-4 New story" and start the review, it will automatically transition the TIS-4 issue from 'In Progress' to 'In Review'. If you reject, abandon or close the review, it will also transition the "TIS-4" issue accordingly.</p>
--	--

### User mapping from the development tools to Jira

The following process describes how a development tool user is mapped to a Jira user for workflow triggers. It applies to all events, however each development tool uses a different email address and username for the mapping (see the bullet point following the process description below).

- **Process:** The user initiating the event in the development tool is mapped to a Jira user by matching the email address, then the username, i.e.
  - *Single Jira user with a matching email address* Transition the issue as the Jira user.
  - *No Jira users with a matching email address* Transition the issue as an anonymous user.
  - *Multiple users with a matching email address in Jira* Try to find a matching username in that group of users. If there is a Jira user with a matching username, transition the issue as the Jira user. If there is no matching username, transition the issue as an anonymous user.
- Email address and username used for user mapping:

Event (s)	Email address and username used for user mapping
All pull request events	The Bitbucket Server email address and username of the user who actioned the pull request.
Commit created	The email address associated with the commit and the Bitbucket Server username that the email address maps to. If the email address does not map to a username, the authors "name" from the commit will be used.
Branch created	The Bitbucket Server email address and username of the authenticated user that pushed the branch to Bitbucket Server.

Event (s)	Email address and username used for user mapping
Commit created	The email address associated with the commit and the Fisheye username that the email address maps to. If the email address does not map to a username, the authors "name" from the commit will be used.
Branch created	This event is not mapped to a Jira user. This means that the issue will be transitioned as an anonymous user.
All review events	The Crucible email address and username of the authenticated user that actioned the review.

Event (s)	Email address and username used for user mapping
All pull request events	The Bitbucket email address and username of the user who actioned the pull request. Note, the Bitbucket user needs to have made at least one commit (with that email address configured for their profile), otherwise the pull request cannot be mapped to a Jira user. This means that the issue will be transitioned as an anonymous user.
Commit created	Email address associated with the commit and the Bitbucket username that the email address maps to. If the email address does not map to a username, the authors "name" from the commit will be used.

Branch created	This event is not mapped to a Jira user. This means that the issue will be transitioned as an anonymous user.
Event (s)	Email address and username used for user mapping
Pull request created / Pull request merged	GitHub email address and username of the user who actioned the pull request. Note, the GitHub user needs to have made at least one commit (with that email address configured for their profile), otherwise the pull request cannot be mapped to a Jira user. This means that the issue will be transitioned as an anonymous user.
Commit created	Email address associated with the commit and the GitHub username that the email address maps to. If the email address does not map to a username, the authors "name" from the commit will be used.
Branch created	This event is not mapped to a Jira user. This means that the issue will be transitioned as an anonymous user.

### Event handling and event limits

In most cases, the processing of events from your development tools into automatic issue transitions should be seamless. However, sometimes there may be delays in issues transitioning or issues not transitioning at all, due to how events are handled or event limits.

- **Event handling** Events are handled differently depending on whether the development tool connects to Jira via the DVCS connector or an application link. This can affect whether events are delayed or lost when Jira is unavailable:

Events from Bitbucket and GitHub are processed via the DVCS connector in Jira. The DVCS connector processes events from Bitbucket and GitHub via two synchronization mechanisms: a webhook-triggered synchronization and a scheduled synchronization.

- **Webhook-triggered synchronization:** the DVCS connector uses webhooks in Bitbucket and GitHub to post data to Jira when an event occurs. This is the standard mechanism for processing events, which means that issues should be automatically transitioned almost immediately after a Bitbucket/GitHub event.
- **Scheduled synchronization:** if Jira cannot be contacted when a Bitbucket/GitHub event occurs, the event is stored by the DVCS connector and sent at the next scheduled synchronization (every 60 minutes, by default). This is a backup mechanism in case the webhook-triggered synchronization fails.

Events from Bitbucket Server and Fisheye/Crucible are processed via the application link. However, Bitbucket Server and Fisheye/Crucible are responsible for ensuring that events are sent, and they send them once at the time that the event occurs. This means that if Jira is unavailable when the events are sent, the events will be lost.

- **Event limits** Event limits are imposed on all of the development tools so that Jira is not overloaded with too many events. Any events sent after the event limit is exceeded are lost. Event limits for each development tool are listed below:
  - **Webhook-triggered synchronization:** 10 branches; 100 commits
  - **Scheduled synchronization:** 600 branches (sync interval in minutes x 10); 6000 commits (sync interval in minutes x 100)

The event limits for scheduled synchronizations can be less than 600 branches and 6000 commits, if the synchronization interval is reduced, but never greater.

10 branches; 100 commits per synchronization

A further constraint that applies on top of the 10 branches and 100 commits limits is a 100,000 issue changed event limit. For example, if 100 commits each reference more than 1000 issue keys, the issue changed limit would be exceeded.

6000 events per synchronization

## How triggers relate to other workflow operations/constraints

When a transition is triggered automatically, it ignores any conditions, validators or permissions configured on the transition.

However, post functions are still executed. You need to be careful that if your post function requires a user, that your transition will not be executed by an anonymous user (see [user mapping section](#) above).

## Troubleshooting

If you are having problems setting up a trigger or getting a trigger to work, follow the steps below to troubleshoot your problem.

- [1. Use the trigger diagnostics](#)
- [2. Check for common problems](#)
- [3. Get help](#)

### 1. Use the trigger diagnostics

Your first step in troubleshooting a trigger is to check the diagnostics for it in Jira. The diagnostics can tell you if there is a problem with the connection to your development tools or whether an issue did not automatically transition as expected.

1. Navigate to the Jira administration console > **Issues** > **Workflows** > Find your workflow and click **View** (**Operations** column)
2. In **Text** mode (not **Diagram** mode), click the desired transition.
3. On the transition screen (**Triggers** tab will be showing), click **View details** for the desired trigger to show the diagnostics information.
  - The 'Trigger sources' section lists problems related to the integration between Jira and your development tools. For example, whether you have the correct type of authentication configured.
  - The 'Transition failures' section lists issues that have failed to automatically transition despite the trigger firing. For example, an anonymous user was mapped to the transition but the transition has a post function that requires a non-anonymous user.

### 2. Check for common problems

If you cannot resolve your problem with the information from the trigger diagnostics, check the list of common problems below for possible causes and solutions.

#### I cannot add a trigger to a transition:

Cause	Solution
Jira or your development tools are not the correct version	Install/Upgrade to the correct version. <i>You must have <b>Jira 6.3.3+</b> and one of the following development tools to enable workflow triggers: <b>Bitbucket Server (Stash 3.2.0+)</b>, <b>Fisheye/Crucible 3.5.2+</b>, <b>Bitbucket</b>, <b>GitHub</b></i>
Your development tools are not connected to Jira correctly	Check the configuration of your connection: <ul style="list-style-type: none"> <li>• <b>Jira + Bitbucket Server/Fisheye/Crucible:</b> You need to configure a two-way application link using OAuth with 2LO and 3LO.</li> <li>• <b>Jira + Bitbucket/GitHub:</b> You need to configure the DVCS connector correctly.</li> </ul> <p>For more details, see <a href="#">Integrating with development tools</a> .</p>
The trigger that you are trying to add has already been added to the transition	Do nothing.  <i>All triggers are unique per transition, that is, you can only add a trigger to a transition once.</i>

**The issue does not transition:**

Cause	Solution
Your project is not using the workflow that has been configured with triggers	Navigate to your project's summary > <b>Administration</b> > <b>Workflows</b> , and check that your project is using the workflow that you have configured with triggers.
You have not saved your workflow changes where the triggers were added	Navigate to the workflow that you added triggers to. Check that it has been published by viewing the workflow transitions and confirming that your triggers are present.
Jira cannot be reached by your DVCS	<p>Wait an hour. If it still cannot be reached after an hour, check that the connection to your DVCS is configured correctly, see <a href="#">Integrating with development tools</a> .</p> <p><i>If triggers are not configured or Jira is not reachable from Bitbucket/GitHub, then the delay might be up to one hour, as there is still an hourly synchronization of commits /branches/pull requests happening regardless of the triggers configuration. For more information, see the <a href="#">Event handling and event limits section</a> above.</i></p>
Your DVCS repository is not linked to the synchronized DVCS account	<p>Navigate to the Jira administration console &gt; <b>Add-ons</b> &gt; <b>DVCS Accounts</b> and enable your repository.</p> <p><i>If you have not configured Bitbucket or GitHub to <a href="#">autolink new repositories</a>, you may have repositories that are not enabled (i.e. linked to your DVCS account). This means that events from the unlinked repository will not be sent to Jira, hence the issue will not transition automatically, even if you have configured a trigger.</i></p>
Your commits are too old	<p>Only commits less than 21 days old will cause a transition. This is to prevent bulk uploads from causing bulk transitions.</p> <p><i>If you want to work around this, you can change the 21 day constraint by editing the <b>ji-ra-config.properties</b> file (in your Jira home directory) and adding the following property:</i></p> <pre data-bbox="402 1451 1123 1478">jira.devstatus.commitcreated.age.timeout=P2D</pre> <p><i>where P2D is an example <a href="#">ISO-8601</a> duration representing 2 days.</i></p>
The operation is not permitted for anonymous users	<p>Check that each user in your development tools maps to a Jira user.</p> <p><i>Certain issue operations will throw exceptions when the transition is performed by an anonymous user. These are:</i></p> <ul data-bbox="427 1720 1327 1814" style="list-style-type: none"> <li>• The <b>CreateIssue</b> event (this probably relates to 'Create' or 'Create Issue' transition in your workflow)</li> <li>• Post functions that assume a user is performing the transition</li> </ul> <p><i>A triggered transition is performed by an anonymous user if the event in the development tool cannot be mapped to a Jira user. For more information, see the section on <a href="#">user mapping</a> above.</i></p>

The maximum number of automatic transitions permitted for an issue has been exceeded	<p>Check that your workflow transitions do not end in an infinite loop.</p> <p><i>By default, only 50 automatic transitions are permitted per issue. This is to prevent issues from becoming stuck in infinite loops. If your workflow actually requires more than 50 automatic transitions per issue, you can override this constraint by editing the <b>jira-config.properties</b> file (in your Jira home directory) and adding/updating the following property:</i></p> <pre>jira.automatic.transitioning.issue.limit</pre>
Automatic issue transition events are incorrectly suppressed by the development tool	<p>Change the repository/project settings to allow events to be sent.</p> <p><i>You may have configured Bitbucket Server ( Stash 3.3 - 3.5) or Fisheye (3.5+) repositories to suppress events sent to Jira for workflow triggers, if duplicate events were being sent. Duplicate repository events may be sent to Jira when you have the same repository indexed by multiple development tools. Note, Jira will automatically remove duplicate commit events (Jira 6.3.3+) and branch creation events (Jira 6.3.11+) when processing workflow triggers.</i></p> <p><i>You shouldn't suppress repository events from Bitbucket Server or Fisheye, unless duplicate events are causing issues to transition incorrectly.</i></p>

**The issue transitions but not as expected:**

Cause	Solution
You have configured a trigger on a global transition	<p>Investigate how the trigger event affects issues in different statuses. Consider removing the trigger from the global transition.</p> <p><i>We recommend that you do not configure triggers for global transitions, unless you are confident that you understand exactly how the trigger will affect the behavior of the issue. See <a href="#">Triggers and global transitions</a> above for more information.</i></p>
Workflow conditions, validators and permissions are intentionally ignored for automatic issue transitions	<p>Do nothing.</p> <p><i>If you were expecting workflow conditions, validators or permissions to be applied to an automatic issue transition, then please note that none of these apply. Related to this, post functions do apply to automatic issue transitions.</i></p>
Your workflow is shared across multiple projects	<p>You may need to copy your workflow, if you want triggers to apply to the workflow for some projects but not others.</p> <p><i>Triggers apply to the workflow. If a workflow is shared across multiple projects, it will include all triggers that have been configured for it.</i></p>
Duplicate automatic issue transition events are being sent by multiple development tools	<p>Change the repository/project settings in one (or more) of your development tools to prevent events from being sent.</p> <p><i>Duplicate repository events may be sent to Jira when you have the same repository indexed by multiple development tools. Jira will automatically remove duplicate commit events (Jira 6.3.3 and later) and branch creation events (Jira 6.3.11 and later).</i></p> <p><i>If you are not using the latest Jira version and have duplicate repository events causing incorrect issue transitions, you can configure Bitbucket Server ( Stash 3.3 - 3.5) and Fisheye (3.5+) repositories to suppress events sent to Jira for workflow triggers.</i></p>

**The information recorded for the transition is not correct:**

Cause	Solution
The users in your development tools do not map to users in Jira	<p>Check that each user in your development tools maps to a Jira user.</p> <p><i>If users are not mapped correctly, then the user for the issue transition will be anonymous. For more information, see the section on <a href="#">user mapping above</a>.</i></p>
Known issue: The correct user is only shown on the 'History' and 'Activity' tabs for issues in Jira, and in notification emails. In other notifications, e.g. 'Transitions' tab for issues, HipChat notifications, etc, an anonymous user is shown.	<p>Do nothing.</p> <p><i>This is a known issue that will be fixed in a future release.</i></p>

### 3. Get help

If you still cannot resolve your problem, there are a number of other help resources available, including our applications forums, [Atlassian Answers](#), and our [support team](#).



#### Do more with Jira

Take your workflows to the next level with these apps from the [Atlassian Marketplace](#):

- [Version Released Workflow Trigger](#): Automate workflow transition on Version release action
- [Automated Release Notes for Jira](#): Set up rules to generate release notes based on your needs via Jira triggers

# Using validators with custom fields

Use the 'Fields Required' workflow validator that is packaged in the [Jira Suite Utilities](#).

Please note the following caveats regarding validation of data by the 'Fields Required' workflow validator at the time of issue creation:

- fields that you set up as "required fields" are not flagged as such in the form to the end-user
- such fields can be cleared at a later time, which is not what you may have intended
- apps will not detect the requirement as implemented by the workflow validator, so may fail later during usage

The reason 3rd party tools are needed is because Jira's interpretation of "required" from a project's field configuration on some custom field means that the field is now required across all screens available to that project, regardless if the screen doesn't actually display that particular field. 3rd party tools, like the Jira Suite Utilities' 'Fields Required' validator, are effectively a more granular means to control fields at the step or screen level at a project, instead of at the project level by the project's field configuration.

# Using XML to create a workflow

Jira's workflow editor generates OSWorkflow XML definition files that are stored in Jira's database. If you need to take advantage of an OSWorkflow-based feature that is not available in Jira's workflow editor, you can define the workflow in XML and then import it into Jira as described below.

Once the XML workflow has been imported, Jira's workflow editor should be able to display most OSWorkflow definitions, even if it does not support creating or editing them.

For example, conditional results of workflow transitions are displayed in the Other tab on the View Workflow Transition page.

 The Other tab is only visible if a transition has elements that the editor does not directly support.

## Importing an XML workflow into Jira

1. Log in as a user with the **Jira System Administrators** [global permission](#).
2. Choose **Administration**  > **Issues**. Select **Workflows** to open the Workflows page, which displays all of the workflows in your system.
3. Click the **Import from XML** button to open the Import Workflow dialog box.
4. In the Name field, type a name (usually 2-3 words) to identify your new workflow.
5. *(Optional)* In the Description field, type a detailed description of your new workflow.
6. For the Workflow Definition option, you can do either of the following:
  - Upload an XML workflow definition file to do this, choose the **Provide a full path to an XML file...** option, and in the File Path field, type the full path to your XML workflow definition file.  
 This path must be local one, so your XML workflow definition file must be located on your Jira server.
  - Paste the contents of an XML workflow definition file into Jira to do this, choose the **Paste the workflow XML definition** option, copy the contents of your XML workflow definition file, and in the Workflow Definition (XML) field, paste this copied content.
7. Click the **Import** button.

## Copying a workflow between systems

Sometimes, it is useful to create a workflow in a test system and then copy it into a production system. To do this:

1. In the test system, export the workflow to XML by clicking the **XML** link next to the workflow in the list shown on the View Workflows page and save the output into a file.
2. In the production system, import the file via the 'import a workflow from XML' link as described above.

### When importing an XML workflow into Jira:

- Jira's XML workflow definitions contain references to Jira meta attributes. For example, the id of the linked Jira status of each workflow step is stored as a 'Jira.status.id' meta attribute in the step's definition. Therefore, when manually creating workflows in XML, please ensure that all referenced external entities exist before you import the workflow into Jira.

### When copying a workflow between systems:

- Please note that conditions, validators and post functions can have parameters that might be valid in one system and not in another. For example, different systems might contain different sets of values for the 'Resolution' field. This would be a problem if the 'Update Issue Field' post function is used to set the 'Resolution' field to a value that exists in one system but not the other.

# Workflow properties

You can use workflow properties to implement restrictions on certain steps or transitions of a workflow ([below](#)).

**⚠ Please Note: Not everything on this page is recommended!**

- We do not recommend using all of these types of workflow properties as we cannot guarantee that some data and operations (e.g. bulk operations) will not be broken. **Hence, use these types of workflow properties at your own risk!**

**i** For details on how to implement workflow properties (i.e. step and transition properties) in your workflow, please refer to [Working with workflows](#).

## Available Jira workflow properties

There are a few workflow properties which you can use in a transition or step of a workflow. Here are some helpful links:

- [Jira API Documentation - JiraWorkflow constant values](#)

Name	Values	Related Issues	References	Notes
<code>jira.field.resolution.exclude</code>	Resolution id			Add comma-separated resolution ids to the transition properties where you want to not show certain resolutions
<code>jira.field.resolution.include</code>	Resolution id	<a href="#">JRA-16443</a>		Add comma-separated resolution ids to the transition properties
<code>jira.i18n.submit</code>	i18n property key	<a href="#">JRA-6798</a>		Transition (usage: action submit button name)
<code>jira.i18n.title</code>	i18n property key	<a href="#">JRA-6798</a>		Transition (usage: action name, etc.)
<code>jira.issue.editable</code>	true, false		<a href="#">Working with workflows</a>	Step
<code>jira.permission.*</code>	user1, user2 / group1, group2 / role / ...?	<a href="#">JRA-6381</a> <a href="#">JRA-34621</a> <a href="#">JRA-35917</a>	<ul style="list-style-type: none"> <li>• <a href="#">WorkflowBasedPermissionManager</a> class description (API documentation)</li> <li>• <a href="#">Permissions based on Workflow Status</a></li> <li>• <a href="#">For link permissions</a></li> <li>• <code>jira.permission.edit.group=jira-administrators</code> means that only Jira administrators can edit an issue (<a href="#">blog</a>)</li> </ul>	Step (usage: only to restrict permission to either roles, group, or users when issue is in that step)

opsbar- sequence	Integer value greater than or equal to 0		<a href="#">Advanced workflow configuration (Customizing transitions)</a>	Transitions on the 'View Issue' page
---------------------	--	--	---	---

# Configuring Jira Service Management approvals

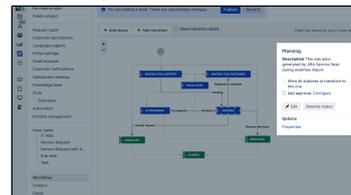
Jira Service Management allows you to add an approval step to a status in a workflow, which allows you to specify if an approval is needed for issue types that are using this workflow. Thanks to that, you can make sure the request is reviewed and approved by the right people before it can progress to the next step.

## How it works

Here's a sample flow that will show you how approvals are added to your workflow, configured, and then displayed on requests:

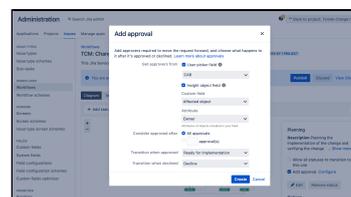
### 1. Adding the approval to your workflow.

First you'll select a workflow step to which you want to add an approval, add it, and move to the configuration screen. You can add approvals to multiple workflow steps.



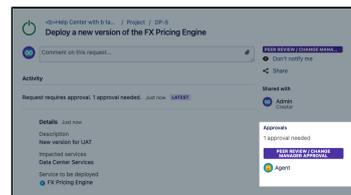
### 2. Configuring your approval.

Next, you'll work out the details of your approval - choose the right custom field that will point to your approvers, and specify the *Approve* and *Decline* transitions, just so your request moves to the right status when it's approved or declined.



### 3. Approving the request.

When a request reaches the approval step, the nominated approvers will be added to your request. They will also receive an email notification.



## Additional information

Here's some additional information to help you better understand how approvals work under the hood:

The option to add an approval step is only available if the workflow is associated with at least one Jira Service Management project. If you add an approval step to the workflow, and that workflow is also used by a non-Jira Service Management project, the issues in the non-Jira Service Management projects can still be worked on in the usual way, but the approval step will *not* be enforced for those issues.

This depends on the number of transitions going out of the approval step on your workflow:

- If the approval step has only two existing transitions, the request can't be transitioned by anyone until the approver has either approved or declined the request. An approval is always related to two transitions - one for *Approved*, and one for *Declined*. Agents will be able to add, edit, or remove approvers, but they won't be able to transition the request using any of these transitions.
- If there are more transitions, the request can be transitioned via any of the ones that aren't used in your approval, without requiring an approval. The whole point of configuring approvals is to not have additional transitions that can omit them, but in some cases you might want to add, for example, *Emergency override* to move the request forward if the usual approvers aren't available.

As mentioned in the previous question, an approval step that has only two transitions makes sure that an approver must make a decision before the request can be progressed. If you want to achieve that, set the approval step on a workflow status that only has two outgoing transitions, one to **Decline** and one to **Approve**.

Yes, you can configure single or multiple approvers, and even require how many of them need to approve your request before it's considered approved. For example, you may want a manager to initially approve a request and then two members of your finance team to make the final approval on the request. You can also have two approval steps on your workflow, each with different approvers.

As for customers and raising requests in the customer portal, you can also decide whether customers can select approvers or if you want them to be selected from a predefined list.

In Jira, many transitions use transition screens that let you modify requests when they move via these transitions, for example by adding a resolution, comment or some extra description.

Transitions that are used in your approvals **omit these screens**, which is important when your *Approve* or *Decline* transitions are going to a final status, like Done. In such a case, you can use post-functions to automatically add modifications that you'd usually add by using a transition screen. In the case of a final status, it's a good idea to add a post function that sets the resolution of the request so it's considered closed by Jira Service Management.

## Types of approvals

When you configure approvals, you select custom fields that in turn point to users who will be nominated as approvers. You can configure two types of custom fields, each will be used for a different type of approvers:

Type	Description
<b>User picker field</b>	<p>This type uses any Jira user picker custom field (single or multiple) to add approvers from users that are configured to display in this field.</p> <p><b>When to use this field</b></p> <ul style="list-style-type: none"> <li>When you want to add approvers from your Jira users or groups. It can be any user or group who is added as approvers will depend on how you configure the custom field.</li> </ul> <p><b>Required configuration</b></p> <ul style="list-style-type: none"> <li>User picker custom field including some users or groups</li> </ul>
<b>Insight object field (Data Center)</b>	<p>This type uses an Insight object custom field to add approvers that are related to your assets from Insight, for example their owners or groups responsible for their maintenance.</p> <p><b>When to use this field</b></p> <ul style="list-style-type: none"> <li>It would typically be used when you're raising a request and include one of your assets in it. People with relations to this asset will be added as approvers.</li> </ul> <p><b>Required configuration</b></p> <ul style="list-style-type: none"> <li>Insight object field mapped to assets in Insight. Supported types are: <ul style="list-style-type: none"> <li>Insight Object/s</li> <li>Insight Object (single) (deprecated)</li> <li>Insight Object (multiple) (deprecated)</li> </ul> </li> <li>Object types and objects (assets) in Insight configured to have a User or Group attribute that includes some users or groups</li> </ul> <div style="border: 1px solid green; padding: 5px; margin-top: 10px;"> <p> If you're looking to use this type, it's best to follow our detailed guide: <a href="#">Adding approvers from Insight to requests in Jira</a>.</p> </div>

## Adding and configuring your approvals

To add and configure approval steps on your workflow:

Approvals are based on custom fields that point to users. You need to create and configure these custom fields in your instance, and also add them to the *Create* and *Edit* screens associated with the workflow. You can use the same field for more than one approval step, but it might get confusing for the approvers, so consider creating separate fields.

The steps will differ based on which type of approvals you're looking to configure.

### Creating a user picker custom field

1. Create a user picker custom field. See [Adding custom fields](#).
2. Associate it with the *Create* and *Edit* screens.

### Creating an Insight object field

 This custom field relies on Insight and additional configuration. If you're new to Insight, it's best to follow our detailed tutorial instead. See [Adding approvers from Insight to requests in Jira](#).

1. Add a User or Group attribute to your object types in Insight. See [Adding attributes to object types](#). If you're choosing a group, try to keep it small as bigger groups can affect performance.
2. Add one of the supported Insight custom fields and map it to your Insight objects (assets). See [Adding Insight custom fields to screens in Jira](#).
3. Associate it with the *Create* and *Edit* screens. See [Adding custom fields](#).

### Before you begin

Jira Service Management request types are mapped to Jira issue types, and the issue types are in turn mapped to a workflow. When you add the approval step to a workflow, it will be applied to all issue types mapped to that workflow, and hence all request types mapped to those issue types. It's important to make sure your approval step is valid for all issue types. If you want to add an approval step for just one request type, you should create a separate workflow and issue type that you can map to the request type individually.

For more information on how to do this, read up on [creating issue types](#), [associating issue types with projects](#), and [working with workflows](#).

### Steps

To add an approval step:

1. Choose **Administration > Projects**, and select the relevant project.
2. In the **Project settings** menu, select **Workflows**. All workflows associated with your project will display.
3. Select **Edit** in the **Actions** column of the workflow you want to modify.

4. Select the status you wish to add the approval step to. The status dialog will display.

The screenshot shows the Jira workflow editor interface. On the left is a sidebar with navigation options like 'Request types', 'Customer permissions', 'Language support', etc. The main area displays a workflow diagram with states: WAITING FOR SUPPORT, WAITING FOR CUSTOMER, ESCALATED, IN PROGRESS, PENDING, CANCELED, and RESOLVED. A 'Planning' status dialog is open on the right, showing a description and options to 'Add approval' or 'Remove status'.

5. Check the **Add approval** check box in the status dialog. The Add approval dialog will display.
6. Choose which custom fields to use, how many approvers are required, and which transitions to use for approval and decline.

The screenshot shows the 'Add approval' dialog box. It prompts the user to 'Add approvers required to move the request forward, and choose what happens to it after it's approved or declined'. The dialog includes fields for 'Get approvers from' (User picker field), 'CAB' (dropdown), 'Insight object field' (checked), 'Affected object' (dropdown), 'Attribute' (Owner), and 'Consider approved after' (All approvals). It also has dropdowns for 'Transition when approved' (Ready for implementation) and 'Transition when declined' (Decline). 'Create' and 'Cancel' buttons are at the bottom.

7. Click **Create** to add your approval step.

Don't forget to click **Publish** to make your workflow available! You should also check that if the transitions you're using for the approval step lead to a status in the Done category, they should have a post function to set the resolution of the request so that it shows as closed.

The following scenarios apply only to Jira user picker fields. In the case of Insight object fields, approvers are always added automatically from the attribute that you select while configuring the approval.

### Let customers select approvers manually

This is useful if you don't know who the approver may be, but the customer does for example, it's their manager. In this scenario, you need to make sure the custom field is available on the customer portal so customers can see and use it.

To add the custom field to the customer portal and show it:

1. In your project, go to **Project settings > Request types**.

2. Select **Edit fields** next to the request type you want to add the field to.
3. Click **+ Add field**, select your field, and click **Apply**.
4. Select the display name, field help, and choose whether the field is required.
5. Click **Update** to save your changes.
6. The field is added as **Shown**. You can change the order of the fields by dragging and dropping.

### Automatically add approvers from a predefined list

This is useful if the customer doesn't know who the approver may be, or if you know there is a set list of approvers for the request. The customer won't need to enter any approvers—they will be automatically added after the request reaches your approval step. In this scenario, you need to enable the custom field on the customer portal, but keep it hidden.

To add the custom field to the customer portal and hide it:

1. In your project, go to **Project settings > Request types**.
2. Select **Edit fields** next to the request type you want to add the field to.
3. Click **+ Add field**, select your field, and click **Apply**.
4. Select the display name, field help, and choose whether the field is required.
5. Click **Update** to save your changes.
6. Click **Hide** next to the field you added. At this point, you will need to populate the list of approvers.

### Let agents select approvers manually

This is useful if the customer doesn't know the approver and the approver depends on the information in your request, so they can't be assigned automatically. In this case, the agent will need to review the request and manually add an approver.

In this scenario, you don't have to do anything as agents can select approvers by default. Agents will be able to see the custom field and edit it in the **People** section of the request. Note that there is also an **Approvals** section that lists all approvers, however you can't make any edits here.

If the field isn't showing, you may need to get a Jira administrator to check the field is still [available on your project screens](#).

# Archiving an issue

By archiving an issue, you can hide it in Jira, but preserve the data it contains in case you need it later. It's good practice to archive *Done* or *Resolved* issues, or those whose resolution due date has passed, so that they don't clutter your Jira instance. For example, many customers archive issues that haven't been updated for the last 2 years.

You can archive one or multiple issues. If you're archiving a large number of issues this may take a while. No reindexing is needed when you archive or restore issues.

 This feature is available for Data Center only.

- [Before you begin](#)
- [Archiving an issue](#)
- [Archiving multiple issues as part of bulk change](#)
- [Archiving via API](#)
- [What happens to an issue after you archive it?](#)
  - [Issues](#)
  - [Index](#)
- [Restoring an issue](#)
- [Browsing archived issues](#)
- [Exporting archived issues](#)
- [Archiving issues and reports](#)
- [Archiving FAQ](#)

## Before you begin

- By default, you must be a Jira Administrator or Jira System Administrator to archive or restore issues. However, an administrator might grant you the global Archive Issues or the Archive Issues for a Project permission, as well as the Restore Issues or the Restore Issues for a Project permissions. The Archive and Restore per project permissions allow you to archive and restore issues in a specific project. By default, these permissions are not enabled.
- You need to be a Jira System Administrator to export archived issues.
- In addition to archive and restore, there are the global Browse Archive and Browse Project Archive permissions. These permissions allow you to access the Archived issues pages and use the filters to find the issues you need - either all of them or those that belong to a specific project.

## Archiving an issue

You don't have to prepare issues in any way before archiving them. You can archive any issue, and restore it later, if needed. To archive an issue:

1. Go to **Issues**.
2. Find the issue you want to archive, open it, and select **More > Archive**.  
The issue will be immediately hidden from view and moved to the archive. Each issue will be archived with all its subtasks.

## Archiving multiple issues as part of bulk change

If you want to archive thousands of issues at once, instead of selecting the issues manually, you can make a bulk change. By default, this option allows you to archive all the issues on the current page or a maximum of 1000 issues. However, a system admin can raise this limit if needed.

You either need to be a Jira admin or a delegated admin to make a bulk change.

1. Open **Issues**.
2. Select the issues you want to view.

3. Open the **Tools** drop-down and select the number of issues you want to perform the change on. You can perform the change on the current page or a maximum of 1000 issues. You might not have these options if you only have a few issues. Then you can only select to perform the operation on the issues you have.
4. Select the issues you want to archive and click **Next**.

 You can also right-click one or several issues on your board or backlog and select **Bulk change**.

5. Select **Archive issues**.
6. Select whether to send a notification about the change to the users involved in the issues.

The issues will be archived with all their subtasks. Note that issues might have multiple subtasks so archiving might take a while because of a great number of subtasks.

Archived issues disappear from the dashboard and search. If you use Jira Software or Jira Service Management, they might also disappear from other places. For more information, see [What happens to my issues](#).

## Archiving via API

You can also use API to archive your issues. See [API documentation](#).

Archiving using REST API can also help you to archive more than 1000 issues at once. For details, see [Easy way to archive a lot of issues](#).

## What happens to an issue after you archive it?

Here's a few things you should know about:

### Issues

- Issues will be read-only and accessible only with a direct link, mentions in other issues or applications, or on the **Archived issues** page (**Issues > Archived issues**). You won't be able to modify them but for audit purposes you can view or, if you're a Jira system admin, export them to a CSV file. To export archived issues with all their data, go to **Issues > Archived issues**, open the **Export** drop-down and click to export all issues or only the selected ones.
- Issues will no longer appear in the list of issues in projects, search results, JQL auto-complete, dashboards, or reports.
- In Jira Software, archived issues will also disappear from the Scrum and Kanban boards and backlogs.
- In Jira Service Management, archived issues will also disappear from the customer portal, queues, and all other places they previously appeared.
- Archived issues are deleted from Jira index and because of that Custom Field Optimizer does not have the full set of information about the issues which use a particular custom field. This can result in some unwanted behaviour such as custom fields not displaying for archived issues and for the issues that have been restored. For more information, see [Jira Knowledge Base](#).

### Index

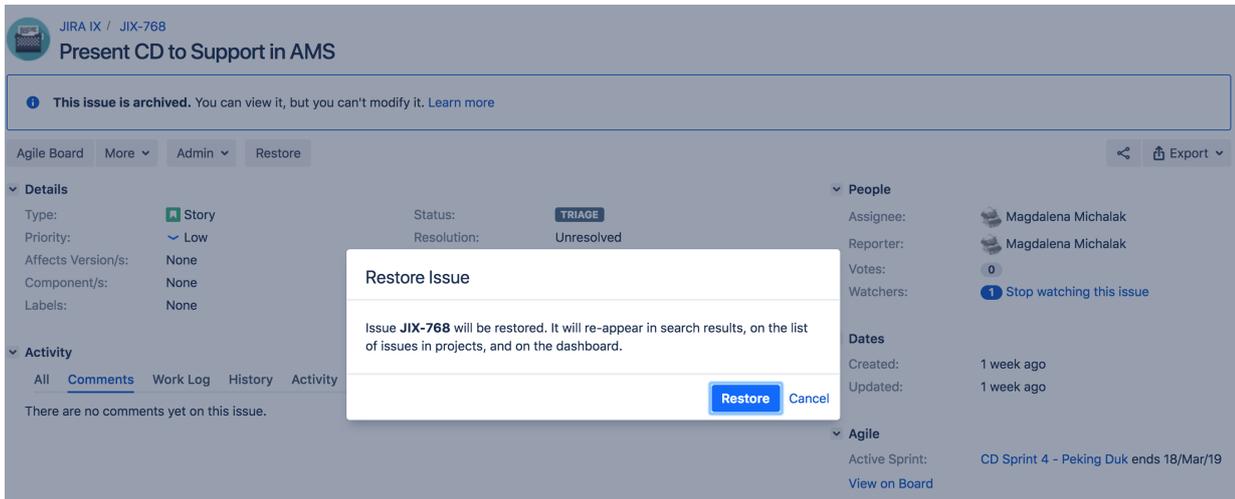
- Issue data will be ignored and removed from the index. This enhances Jira performance because Jira stores less data.

## Restoring an issue

All the issue data remains in the database, so you can restore it whenever it's needed again. Currently, you can restore one issue at a time.

If you want to restore multiple issues, do that using the REST API.

To restore an archived issue, open it with the direct link or go to **Issues > Archived issues**, filter to find the issue and click the **Restore** button.



JIRA IX / JIX-768  
Present CD to Support in AMS

**This issue is archived.** You can view it, but you can't modify it. [Learn more](#)

Agile Board More Admin Restore

**Details**

Type: Story  
Priority: Low  
Affects Version/s: None  
Component/s: None  
Labels: None

Status: TRIAGE  
Resolution: Unresolved

**Restore Issue**

Issue **JIX-768** will be restored. It will re-appear in search results, on the list of issues in projects, and on the dashboard.

Restore Cancel

**People**

Assignee: Magdalena Michalak  
Reporter: Magdalena Michalak  
Votes: 0  
Watchers: Stop watching this issue

**Dates**

Created: 1 week ago  
Updated: 1 week ago

**Agile**

Active Sprint: CD Sprint 4 - Peking Duk ends 18/Mar/19  
[View on Board](#)

**Activity**

All Comments Work Log History Activity

There are no comments yet on this issue.

The issue will be restored to its original state, and brought back to Jira. While issues are being restored Jira is also reindexed so that no additional action is needed for the issues to be visible and searchable again. The issue will be restored with all its subtasks.

### **i** Restore issue from an archived project

If an issue has been archived together with the project it belonged to, you need to restore the project to restore the issue. For restoring projects, see [Archiving a project](#).

## Browsing archived issues

**i** To browse all archived issues you must be a [Jira System Administrator](#) or have the Browse Archive permission. To browse issues from a specific project you must have the Browse Project Archive permission.

To browse issues go to **Issues > Archived issues** and apply any of the filters available.

**Archived issues**

Browse through the list below to find the archived issues you need. The list contains both: issues from archived projects and archived issues from active projects.

Filter by: Archived date: All ▾ Archiver: All ▾ Project: All ▾ Type: All ▾ Reporter: All ▾ [Search](#)

Showing results 1-50 of 50 ↻

T	Archived date	Archiver	Project	Key	Summary	Reporter	Status	Resolved	Created	Assignee
<input checked="" type="checkbox"/>	24/May/2019	Living Tribunal	Spike	SPK-112	Is there life on Mars?	Shinichirō Watanabe	DONE		24/May/2015	
<input type="checkbox"/>	24/May/2019	Living Tribunal	Time Archived	TIME-12	Cats on Mars	H. George Wells	DONE		24/May/2015	
<input type="checkbox"/>	24/May/2019	Living Tribunal	Odyssey Archived	ODSY-101	Find an efficient way to stop	Dave Bowman	DONE		24/May/2015	
<input checked="" type="checkbox"/>	24/May/2019	Living Tribunal	Bebop	BBP-2312	Bug fixes for the infinity glove	Adam Warlock	DONE		24/May/2015	
<input checked="" type="checkbox"/>	24/May/2019	Living Tribunal	Spike	SPK-112	Is there life on Mars?	Shinichirō Watanabe	DONE		24/May/2015	
<input type="checkbox"/>	24/May/2019	Living Tribunal	Timely	TIME-12	Cats on Mars	H. George Wells	DONE		24/May/2015	
<input type="checkbox"/>	24/May/2019	Living Tribunal	Bebop	BBP-121	Find an efficient way to stop	Dave Bowman	DONE		24/May/2015	
<input checked="" type="checkbox"/>	24/May/2019	Living Tribunal	Bebop	BBP-2312	Bug fixes for the infinity glove	Adam Warlock	DONE		24/May/2015	
<input checked="" type="checkbox"/>	24/May/2019	Living Tribunal	Bebop	BBP-230	Is there life on Mars?	Shinichirō Watanabe	DONE		24/May/2015	
<input type="checkbox"/>	24/May/2019	Living Tribunal	Bebop	BBP-231	Cats on Mars	H. George Wells	DONE		24/May/2015	
<input type="checkbox"/>	24/May/2019	Living Tribunal	Bebop	BBP-232	Find an efficient way to stop	Dave Bowman	DONE		24/May/2015	
<input checked="" type="checkbox"/>	24/May/2019	Living Tribunal	Bebop	BBP-233	Bug fixes for the infinity glove	Adam Warlock	DONE		24/May/2015	
<input checked="" type="checkbox"/>	24/May/2019	Living Tribunal	Bebop	BBP-234	Is there life on Mars?	Shinichirō Watanabe	DONE		24/May/2015	
<input type="checkbox"/>	24/May/2019	Living Tribunal	Bebop	BBP-123	Cats on Mars	H. George Wells	DONE		24/May/2015	
<input type="checkbox"/>	24/May/2019	Living Tribunal	Bebop	BBP-2309	Find an efficient way to stop	Dave Bowman	DONE		24/May/2015	
<input checked="" type="checkbox"/>	24/May/2019	Living Tribunal	Bebop	BBP-2212	Bug fixes for the infinity glove	Adam Warlock	DONE		24/May/2015	

Showing results 1-50 of 50 ↻

Atlassian JIRA Project Management Software • [About JIRA](#) • [Report a problem](#)



If your search matches more than 1000 issues, we will only be able to see the latest 1000 archived issues. In such case, you can either narrow down your search or ask your system administrator to export a complete list of archived issues.

## Exporting archived issues

**i** To export the list of archived issues, you must be a [Jira System Administrator](#).

You can export archived issues to a CSV file for audit purposes. You can choose to export the filtered results or export all archived issues. Then, the file contains all archived issues both archived individually and archived together with the project they belonged to, and sorted by the date of archiving. Each exported issue contains the following fields: system fields, archived by, archive date, and the URL, which is the direct link to the issue.

The filters available allow you to limit the number of exported issues so that they are easier to read in your export file.

You can export your archived issues to CSV by going to **Issues > Archived issues** and click the **Export** down to select your export option.

The screenshot shows the 'Archived issues' page in Jira. At the top, there are navigation tabs for Dashboards, Projects, Issues, Boards, and a 'Create' button. A search bar is on the right. Below the header, there's a description: 'Browse through the list below to find the archived issues you need. The list contains both: issues from archived projects and archived issues from active projects.' Below this are filter options: 'Filter by: Archived date: All, Archiver: All, Project: All, Type: All, Reporter: All' and a 'Search' button. On the right, an 'Export' dropdown menu is open, showing 'CSV (All)' and 'CSV (Filtered results)'. The main content is a table of issues with columns: T, Archived date, Archiver, Project, Key, Summary, Reporter, Status, Resolved, Created, and Assignee. The table shows 15 rows of issues, all with a status of 'DONE' and a creation date of '24/May/2015'. Below the table, it says 'Showing results 1-50 of 50'.



If you select to export all issues, we let you know how many issues there are to export in the confirmation pop-up.

The screenshot shows the 'Export archived issues - CSV (All)' dialog box. The dialog box has a title bar and contains the text '815 issues to export'. There are 'Confirm' and 'Cancel' buttons. In the background, the 'Archived Issues' page is visible, showing a table with columns: T, Archived date, Archived by, Project, Key, Summary, Reporter, Status, Resolved, Created, and Assignee. One row is visible with the following data: T: [checked], Archived date: 19/Jul/19, Archived by: admin, Project: dd, Key: DD-5, Summary: Keyboard shortcuts, Reporter: admin, Status: TO DO, Resolved: [empty], Created: 19/Jul/19, Assignee: admin.

✔ Exporting issues can also be useful to quickly trace the issues that might have been archived by mistake.

Alternatively, you can use REST API. If you want to export issues using the REST API run the following command:

```
/rest/internal/2/archiving?all=true
```

### Archiving issues and reports

Archived issues are removed from index and, as such, they are not picked up by JQL queries used in reports. That is why archived issues are not displayed in reports. For example, if you've created 30 issues in May and archived them later on, and then you run a report showing Created and Resolved issues for that month, the report will not display any results. On the other hand, when you use the archiving feature, reports tend to be faster because archiving decreases the size of the index.

That is why we advise archiving old issues which will not be used in any reporting. For example, if you do not generate reports for a period of time longer than one year, then you can safely archive issues older than that. This way, archived issues will not distort any report metric.

## Archiving FAQ

**Q:** I want to restore subtasks but not an issue. Is this possible?

**A:** No. If you want to restore subtasks you also need to restore the issue these subtasks belong to.

**Q:** I want to delete a project that has some archived issues in it. What happens to these issues?

**A:** The issues get deleted and are no longer stored as archived issues.

**Q: Is the information about archived issues included in batch notifications?**

**A:** Yes. We inform you about every issue archived and restored.

**Q: If I archive an issue and then archive a project this issue belongs to, is it enough if I unarchive this issue to see it?**

**A:** No. You first need to restore the project and then the issue to see it.

# Archiving a project

By archiving a project, you can remove it from Jira and preserve the data it contains in case you need it later. Its good practice to archive inactive or completed projects so they dont clutter your Jira instance. Fewer projects may also mean better performance.

**i** Project archiving is **only available for Jira Software Data Center and Jira Service Management Data Center**. To archive projects in Jira Server, see [this knowledge base article](#).

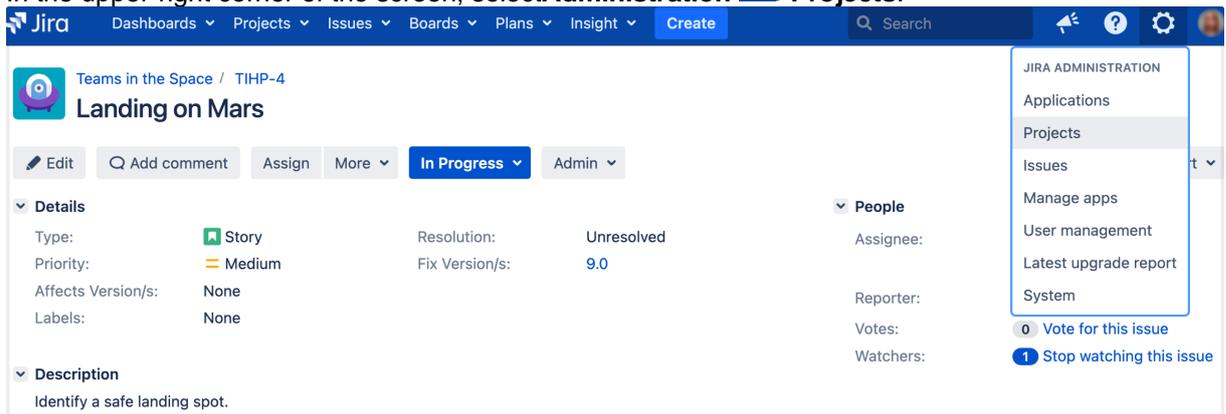
## Before you begin

- You must have the Jira Administrator or Jira System Administrator [global permission](#) to archive or restore projects. For more information about different types of permissions that can be set up in Jira, see [Permissions overview](#).
- You don't have to prepare a project in any way before archiving it. You can archive any project, and restore it later, if needed.

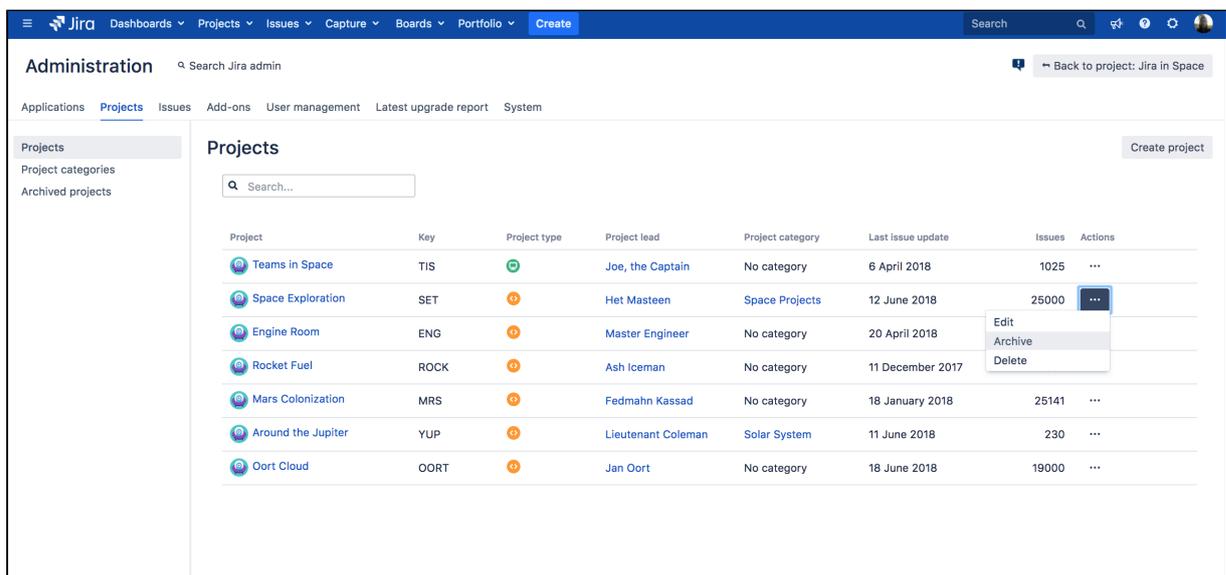
## Archiving a project

To archive a project:

1. In the upper-right corner of the screen, select **Administration**  **Projects**.



2. Find the project that you want to archive and select **Actions** (  ) **Archive**.



The project will be immediately hidden from view and moved to the **Archived projects** page.

3. [Re-index Jira](#) to remove the project's data from the index. You can do it after archiving multiple projects or whenever you see fit. Re-indexing is needed to get the full performance benefits.

### What happens to a project after you archive it?

Here's a few things you should know about:

<b>Project</b>	<ul style="list-style-type: none"> <li>• The project will be restricted, and can't be viewed by anyone.</li> <li>• The project will no longer appear in project pickers, list of projects, search results, or JQL auto-complete. It won't be visible anywhere besides the <i>Archived projects</i> page.</li> <li>• Links to the project will still work.</li> </ul>
<b>Issues</b>	<ul style="list-style-type: none"> <li>• Similarly to the project itself, issues will no longer appear in search results.</li> <li>• Issues will become read-only. You won't be able to modify them, but you can still view them either through direct links or mentions in other projects or applications.</li> </ul>
<b>Customer portal</b>	<div data-bbox="384 976 1430 1055" style="border: 1px solid #ccc; border-radius: 5px; padding: 5px; margin-bottom: 10px;">  Only applicable if Jira Service Management is installed and licensed.         </div> <ul style="list-style-type: none"> <li>• The customer portal will be restricted, and can't be viewed by anyone.</li> <li>• The customer portal will no longer appear in the help center navigation. If you access a customer portal link, you will receive a message advising you that the service project does not exist.</li> </ul>
<b>Configuration</b>	<ul style="list-style-type: none"> <li>• Archiving a project won't affect the configuration it uses (schemes, screens, workflows, custom fields etc.) The configuration remains active and is still shared with other projects. If you change it, the changes will also apply to an archived project once it's restored.</li> </ul>
<b>Index</b>	<ul style="list-style-type: none"> <li>• Project data will be ignored in the index, and removed completely once you re-index Jira. This improves performance by removing data that is stored directly in Jira.</li> </ul>

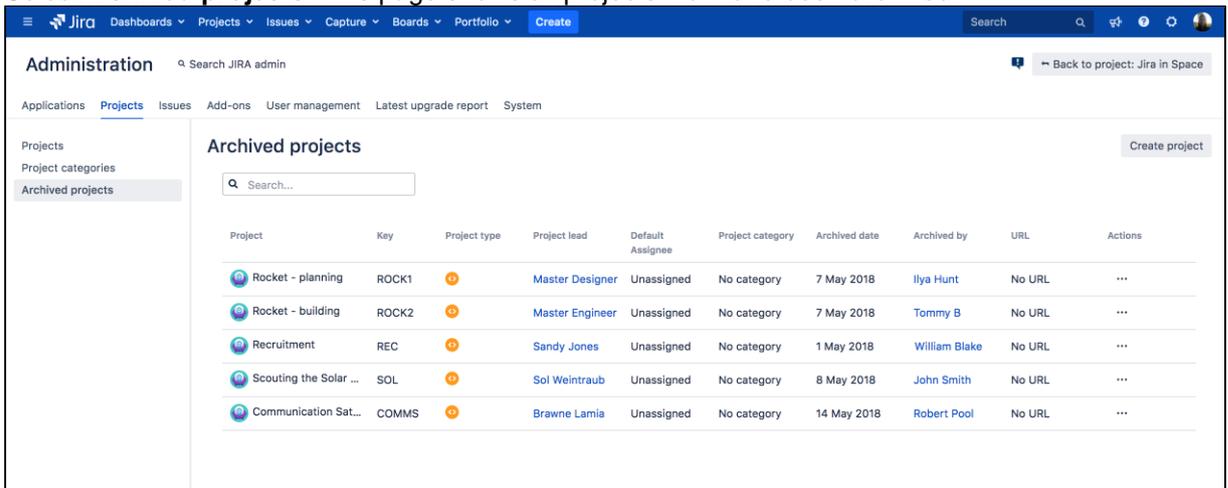
### Restoring and re-indexing a project

All the project data remains in the database, so you can restore it whenever it's needed again.

To restore a project:

1. In the upper-right corner of the screen, select **Administration**  **Projects**.

2. Select **Archived projects**. This page shows all projects that have been archived.



The screenshot shows the Jira Administration interface. The 'Archived projects' section is active, displaying a table of archived projects. The table has the following columns: Project, Key, Project type, Project lead, Default Assignee, Project category, Archived date, Archived by, URL, and Actions. The data rows are as follows:

Project	Key	Project type	Project lead	Default Assignee	Project category	Archived date	Archived by	URL	Actions
Rocket - planning	ROCK1	🔴	Master Designer	Unassigned	No category	7 May 2018	Ilya Hunt	No URL	...
Rocket - building	ROCK2	🔴	Master Engineer	Unassigned	No category	7 May 2018	Tommy B	No URL	...
Recruitment	REC	🔴	Sandy Jones	Unassigned	No category	1 May 2018	William Blake	No URL	...
Scouting the Solar ...	SOL	🔴	Sol Weintraub	Unassigned	No category	8 May 2018	John Smith	No URL	...
Communication Sat...	COMMS	🔴	Brawne Lamia	Unassigned	No category	14 May 2018	Robert Pool	No URL	...

3. Find the project you want to restore, and select **Actions** (⋮) > **Restore**.  
The project will be restored to its original state, and brought back to the list of projects and project pickers.
4. Re-index the project, so its issues appear again in search results.
  - a. Open the project you just restored.
  - b. In the bottom-right corner of the screen, select **Project settings** (⚙️).
  - c. Under the **Project settings** (the left-side panel), select **Re-index project**.
  - d. Select **Start project re-index**.

# Importing and exporting data

At times, you may need to import or export data to or from Jira. You may want to import data from another tool (like Github or Fogbugz), another Jira instance, or from a manually prepared file such as a CSV or JSON file. You may want to export your data so that you can perform some manual manipulation on it, or to move a project from one instance to another. This section of the documentation explains how to perform imports and exports of your data. If you'd like more information on [backing up your data](#), and [restoring a backup](#), please refer to the [System administration](#) section of the documentation.

Search the topics in 'Importing and exporting data':

## Migrating data from other tools

[Learn more](#) about how to import data from various tools. We also have information on how to structure CSV or JSON files for import.

## Moving or archiving projects

[Learn more](#) about how you can move or archive individual or multiple projects between Jira instances.

## Importing to a Cloud instance

[Learn more](#) about moving your data from your server site to a cloud site.

# Migrating from other issue trackers

When migrating from another issue tracking application to Jira, you may wish to take your data with you. You can do it by importing the data into Jira in CSV or JSON format.

 Our website highlights some top reasons why people [migrate from other issue trackers to Jira](#).

- [Importing data from JSON](#)
- [Importing data from CSV](#)

There is also a workaround for [importing comments](#)

# Importing data from CSV

The Jira Importers plugin, which is bundled with Jira, allows you to import your data from a comma-separated value (CSV) file. This might be helpful when you are migrating from an external issue tracker to Jira.

CSV files are text files representing tabulated data and are supported by most applications that handle tabulated data (for example, Microsoft Excel, databases, etc.).

The CSV import feature allows you to import issues from an external (issue tracking) system that can export its data in a structured or tabulated format (preferably CSV).

 Our main website highlights some top reasons why people [migrate from such an external issue tracking system to Jira](#).

## The CSV import process consists of the following stages:

1. Preparing your CSV file (find the instructions [below](#)).
2. Running the CSV file import wizard (find the instructions [below](#)).
  - You can choose to map individual fields and field values during the import process.
  - At the end of the CSV file import wizard, you can choose to create a CSV configuration file that contains the settings you configured while running through the CSV file import wizard. This is useful if you need to test your CSV file import on a test Jira server first before performing the import on a production system.

 Several methods are available for importing data from other issue tracking systems into Jira. Depending on your other issue tracking system, it may be more appropriate to use a different import method instead of exporting data from that system to a CSV file and then importing that CSV file to Jira.

If your other issue tracking system is listed on the [Migrating from other issue trackers](#) page, try using the appropriate method to import data to Jira.

You may also want to use the Bulk issues import feature instead of External system import. Learn more about Bulk issues import in [here](#).

## On this page:

- [Preparing your CSV file](#)
  - [Encapsulating Jira data structure in your CSV file](#)
- [Running the CSV file import wizard](#)
- [Tips for importing CSV data into Jira fields](#)
- [Importing issues in bulk](#)

## Preparing your CSV file

The Jira Importers plugin assumes that your CSV file is based off a default Microsoft Excel-styled CSV file:

- Fields are separated by commas.
- Any content that must be treated literally, such as commas and new lines/"carriage returns" themselves are enclosed in quotation marks.

 For Microsoft Excel and OpenOffice, it is not necessary to quote values in cells as these applications handle this automatically.

## CSV file requirements

In addition to being "well-formed", CSV files have the following requirements.

Each CSV file must possess a heading row with a Summary column

The CSV file import wizard (see more details [below](#)) uses a CSV file's header row to determine how to map data from the CSV file's 2nd row and beyond to fields in Jira.

The header row shouldn't contain any punctuation, except for the commas separating each column. Otherwise, the importer may not work correctly.

The header row must contain a column for the issue's "Summary" data.

Commas (as column/field separators) cannot be omitted

For example, the following format is valid:

```
Summary, Assignee, Reporter, Issue Type, Description, Priority
"Test issue", admin, admin, 1, ,
```

While this one is not valid:

```
Summary, Assignee, Reporter, Issue Type, Description, Priority
"Test issue", admin, admin, 1
```

## Encapsulating Jira data structure in your CSV file

In this section, you'll find solutions for the following issues:

- [Capturing data that spans multiple lines](#)
- [Treating special characters literally](#)
- [Aggregating multiple values into single Jira fields](#)
- [Importing attachments](#)
- [Creating sub-tasks](#)
- [Importing issues into multiple Jira projects](#)
- [Handle unresolved issues](#)
- [Importing worklog entries](#)
- [Importing to multi-select custom fields](#)
- [Importing cascading choice custom fields](#)

### Capturing data that spans multiple lines

Use double-quote marks (") in your CSV file to capture data that spans multiple lines. For example, during import, Jira will treat the following as a valid CSV file with a single record:

```
Summary, Description, Status
"Login fails", "This is on
a new line", Open
```

### Treating special characters literally

Use double-quote marks (") around a section of text to treat any special characters in that section literally. Once this data is imported into Jira, these special characters will be stored as part of Jira's field data. Examples of special characters include carriage returns/enter characters, commas, etc.

To treat a double quote mark literally, you can "escape" them with another double quote mark character. For example:

- Your CSV file might contain the value like "Clicking the ""Add"" button results in a page not found error".
- Once imported, it will be stored in Jira as Clicking the "Add" button results in a page not found error.

### Aggregating multiple values into single Jira fields

You can import multiple values into a Jira field that accepts multiple values. For example, **Fix (for) Version**, **Affects Version**, **Component**, or **Labels**. To do this, your CSV file must specify the same column name for each value you wish to aggregate into the mapped Jira field. The number of column names specified must match the maximum number of values to be aggregated into the mapped field. For example:

```
IssueType, Summary, FixVersion, FixVersion, FixVersion, Component, Component
bug, "First issue", v1, , , Component1,
bug, "Second issue", v2, , , Component1, Component2
bug, "Third issue", v1, v2, v3, Component1,
```

In this example, the **Component** field of the second issue and the **Fix Version** field of the third issue will generate multiple values in appropriate Jira fields upon import.

**i** Be aware that only a limited number of Jira fields support multiple values. The CSV importer will not allow you to import aggregated data into Jira fields that only support a single value.

### Importing attachments

You can attach files to issues, created from your CSV file. To do this, specify the URL of your attachment in an "Attachments" column in your CSV file.

```
Assignee, Summary, Description, Attachment, Comment
Admin, "Issue demonstrating the CSV attachment import", "Please check the attached image below.",
"https://jira-server:8080/secure/attachment/image-name.png", "01/01/2012 10:10;Admin; This comment works"
Admin, "CSV attachment import with timestamp,author and filename", "Please check the attached image
below.", "01/01/2012 13:10;Admin;image.png;file://image-name.png", "01/01/2012 10:10;Admin; This comment
works"
```

**i** URLs for attachments support the HTTP and HTTPS protocols and can be any location that your Jira server must be able to access. You can also use the FILE protocol to access files in the `import/attachmentssubdirectory` of your [Jira home directory](#).

### Creating sub-tasks

You can create sub-tasks of issues through a CSV file import by encapsulating this structure in your CSV file. To do this:

- The CSV file must have two additional columns whose headings should be named similarly to **Issue Id** and **Parent Id**.
- Ensure that each regular (non sub-task) issue is given a unique (sequential) number in the **Issue Id** column. Do not include any value in the **Parent Id** fields for regular issues.
- To create a sub-task of a regular issue in your CSV file, reference the unique **Issue Id** number of the regular issue in the **Parent Id** column. Don't set any value in the **Issue Id** fields for sub-tasks.

For example:

```
IssueType, Summary, FixVersion, FixVersion, FixVersion, Component, Component, Issue ID, Parent ID,
Reporter
Bug, "First issue", v1, , , Component1, , 1, , jbloggs
Bug, "Second issue", v2, , , Component1, Component2, 2, , fferdinando
Bug, "Third issue", v1, v2, v3, Component1, , 3, , fferdinando
Sub-task, "Fourth issue", v1, v2, , Component2, , , 2, jbloggs
```

In this example, the fourth issue will be imported as a sub-task of the second issue, assuming you match the "Issue ID" and "Parent ID" fields in your CSV file to the **Issue Id** and **Parent Id** Jira fields respectively during the [CSV file import wizard](#).

### Importing issues into multiple Jira projects

You can import issues from your CSV file into different Jira projects through a CSV file import. To do this:

- The CSV file must have two additional columns whose headings should be named similarly to **Project Name** and **Project Key**.
- Ensure that every issue represented in your CSV file contains the appropriate name and key in those columns for the Jira projects to which they will be imported.

 The project name and key data is the minimum Jira project data required for importing issues from a CSV file into specific Jira projects.

```
IssueType, Summary, Project Name, Project Key
bug, "First issue", Sample, SAMP
bug, "Second issue", Sample, SAMP
task, "Third issue", Example, EXAM
```

In this example, the first and second issues will be imported into the "Sample" project (with project key "SAMP") and the third issue will be imported into the "Example" project (with project key "EXAM") , assuming you match the "Project Name" and "Project Key" fields in your CSV file to the **Project name** and **Project key** Jira fields respectively during the [CSV file import wizard](#).

### Handle unresolved issues

For fields mapping to Resolution, Priority, and Issue Type, you will get a select list with the available values in Jira. In addition, you can quickly create values that do not exist in Jira by clicking the green plus symbols.

For fields mapping to Status, you will get the select list with Jira's available values, but no plus symbol for creating new status values.

For these four fields, there are two special options in the select list in addition to Jira's available values:

- "Import as blank". If selected, the Jira value to be blank for that field. Note that if you are importing Unresolved issues, you should create a field mapping for the Resolution field and set the value "Unresolved" to "Import as blank".
- "No mapping". This attempts to import the value in the CSV file as-is. Note that using "No mapping" for a field value will result in a failed import if the value is not valid for that Jira field. For fields mapping to Status and Issue Type, default values are used when the "Import as blank" option is selected.

### Importing worklog entries

Your CSV file can contain worklog entries. For example:

```
Summary,Worklog
Only time spent (one hour),3600
With a date and an author,2012-02-10 12:30:10;wseliga;120
With an additional comment,Testing took me 3 days;2012-02-10 12:30:10;wseliga;259200
```

To track time spent, you need to use seconds.

### Importing to multi-select custom fields

Your CSV file can contain multiple entries for the one Multi Select Custom Field. For example:

```
Summary,Multi Select,Multi Select,Multi Select
Sample issue,Value 1,Value 2,Value 3
```

This will populate the Multi Select Custom Field with multiple values.

### Importing cascading choice custom fields

You can import values to a cascading choice custom field using the following syntax:

Summary, My Cascading Custom Field  
 Example Summary, Parent Value -> Child Value

The '->' separator allows you to import the hierarchy.

**i** Currently, Jira doesn't support importing multi-level cascading select fields via CSV ( [JRASERVER-34202](#) - Allow CSV import to support Multi-Level Cascading Select plugin fields **GATHERING INTEREST** ).

## Updating existing issues

From version 4.3 of Jira Importers plugin, you can update existing issues. Your CSV file needs to contain a column that will be mapped to Issue Key during the import. If an issue exists for a given key, it will be updated. For example:

```
issue key,summary,votes,labels,labels
TT-1,Original summary,1,label1,label2
TT-1,,7,label-1,label-2
TT-1,Changed summary,,,
TT-2,Original summary 2,1,label-1,label-2
TT-2,,<<!clear!>>,<<!clear!>>,
```

The first row will create an issue, the second row will set votes to 7 and add two labels. The following row will change the summary. Issue TT-2 will be created with two labels, but the second row will remove those labels with a special marker <<!clear!>>.

**i** Importing a CSV to update existing issues will **reset columns to their default values** if they are not specified in the CSV.

## Running the CSV file import wizard

Before you begin, you need to [back up](#) your Jira data.

To use the External system import feature to import your issues to Jira:

1. Log in to Jira as a user with the [Jira Administrators global permission](#).
2. Select **Administration > System > Import & Export > External System Import**.
3. Select **CSV** to open the **CSV File import** page.

4. On the **CSV File import** page, select your **CSV Source File**. If you want to change the file's encoding and CSV delimiter format, select the **Advanced** heading to reveal this option.

-  • The file will be imported using the **File encoding** type you specify here. The default file type is **UTF-8**.
- If your CSV file uses a different separator character other than a comma, specify that character in the **CSV Delimiter** field.

5. Leave the **Use an existing configuration file** checkbox cleared if you do not have a configuration file or if you want to create a new configuration file. Configuration files specify a mapping between column names in your CSV file's header row and fields in your Jira installation.

-  • If you select this option, you will be asked to specify an **Existing Configuration File**.
- If you do not select this option, then at the end of the CSV file import wizard, Jira will create a configuration file which you can use for subsequent CSV imports (at this step of the CSV file import wizard).

6. Select **Next** to proceed to the **Setup project mappings** step of the CSV file import wizard.
7. On the **Setup project mappings** page, you can either import *all* your issues into either one Jira project (new or existing), or multiple Jira projects. If you choose to import to multiple projects, ensure that your CSV file includes the minimum Jira project data required, which is Jira project name and key. Complete the following fields/options:

<b>Import to Jira Project</b>	<p>Choose one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Select a project.</b> Use this option to import issues to an existing Jira project or create a new project and import issues to it. <ul style="list-style-type: none"> <li>• Start typing the name (or key) of a project that already exists in Jira or use the dropdown menu to select an existing Jira project.</li> <li>• Select <b>Create New</b> from the dropdown menu and in the resulting <b>Add A New Project</b> dialog box, fill in the following fields: <ol style="list-style-type: none"> <li>a. Enter the project <b>Name</b></li> <li>b. Enter the project <b>Key</b></li> </ol> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;">  This will be used as the prefix for all issue IDs in your Jira project. </div> </li> <li>c. Specify the <b>Project Lead</b>.</li> </ul> </li> <li>• <b>Defined in CSV.</b> Use this option to import issues to multiple Jira projects. Ensure that every issue in your CSV file includes data for the Jira <b>Project Name</b> and <b>Project Key</b>. See <a href="#">Importing issues into multiple Jira projects</a> for details.</li> </ul>
<b>E-mail Suffix for New Users</b>	Enter the email address domain for any new users specified in the CSV file that will be added to Jira during the import.
<b>Date format in import file</b>	Specify the date format used in your CSV file. Use the syntax that complies with the <a href="#">Java SimpleDateFormat</a> .

8. Select **Next** to proceed to the **Setup field mappings** step of the CSV file import wizard.
9. In the **Setup field mappings** page, map the fields in the CSV file to the issue fields in the selected project. In the **Jira field** column, select the Jira field that you want to map to the field from the CSV file. For more information about matching CSV fields to Jira fields, see [Tips for importing CSV data into Jira fields](#) below.

**i**

- The **Summary** field must be specified for one of your Jira fields and the **Next** button will remain unavailable until you do so.
- If your CSV file contains more than one of the same field name specified in its header row, the CSV file import wizard will aggregate these into a single field, which will be marked by an  icon at this step of the wizard.
- For CSV fields that have been aggregated by the CSV file import wizard, you will only be able to select Jira Fields that support multiple values.
- If you are importing sub-tasks, remember to match the **Issue ID** and **Parent ID** fields in Jirato those in your CSV file.
- If you are importing issues into multiple projects, ensure that you selected **Defined in CSV** during the **Setup project mappings** step. Remember to match the **Project Name** and **Project Key** fields in Jira to those in your CSV file.

10. To modify the values of any fields' data in the CSV file *before* they are imported into Jira, select the **Map field value** checkboxes next to the appropriate fields.
11. Select **Next** to proceed to the **Setup value mappings** step of the CSV file import wizard.
12. On the **Setup value mappings** page, specify the Jira field values for each CSV file field value that has been detected by the CSV file import wizard.

**i**

- Any fields which **Map field value** checkboxes were selected in the previous step of the CSV file import wizard will be presented on this page.
- Leave a field cleared or clear any content within it if you wish to import the value "as is".
- You can create new **Priority**, **Resolution**, and **Issue Type** values in Jira (i.e. based on the data in your CSV file) by selecting the **Add new** link next to the appropriate field.
- If you are importing a username-based CSV field (e.g. **Reporter** or **Assignee**) and you didn't select the **Map field value** checkbox for this field in the previous step of the CSV file import wizard, then the importer will automatically map imported usernames from the CSV file to (lowercase) Jira usernames.

Regardless of whether or not you select the **Map field value** checkbox, Jira will automatically create usernames based on the data in your CSV file if they haven't been defined in Jira yet.

13. Select the **Begin Import** button when you are ready to begin importing your CSV data into Jira. The importer will display updates as the import progresses, then a success message when the import is complete.

**i**

- If you experience problems with the import (or you are curious), select the **download a detailed log** link to view detailed information about the CSV file import process.
- If you need to import another CSV file with the same (or similar) settings to what you used through this procedure, select the **save the configuration** link to download a CSV configuration file, which you can use at the [first step](#) of the CSV file import wizard.

Congratulations, you have successfully imported your CSV data into Jira! If you have any questions or encounter any problems, please contact [Atlassian support](#).

### Tips for importing CSV data into Jira fields

The following are some helpful tips when importing data from your CSV file into specific Jira fields.

Jira Field	Import Notes
Project	CSV data is imported on a per-project basis. You can either specify an existing Jira project(s) as the target or the importer will automatically create a new project(s) for you at time of import.
Summary	This is the only required field.

Issue Key	You can set the issue key for an imported issue. If an issue with a given key already exists in Jira, it will be updated instead.
Component(s)	You can import issues with multiple components by entering each component in a separate column.
Affects Version(s)	You can import issues with multiple 'Affects Versions' by entering each version in a separate column.
Fix Version(s)	You can import issues with multiple 'Fix Versions' by entering each version in a separate column.
Comment Body	You can import issues with multiple comments by entering each comment in a separate column.
Date Created	Please use the date format specified on the second step of the CSV import wizard.
Date Modified	Please use the date format specified on the second step of the CSV import wizard.
Due Date	Please use the date format specified on the second step of the CSV import wizard.
Issue Type	If not specified in your CSV file, imported issues will be given the default (i.e. first) Issue Type as specified in your Jira system <a href="#">Defining issue type field values</a> . You can also create new Jira values on-the-fly during the import process.
Labels	Import issues with multiple labels by: <ul style="list-style-type: none"> <li>entering each label in a separate column or</li> <li>putting all labels in one column, delimited by a space</li> </ul>
Priority	If not specified in your CSV file, imported issues will be given the default (i.e. first) Priority as specified in your Jira system <a href="#">Defining priority field values</a> . You can also create new Jira values on-the-fly during the import process.
Resolution	If not specified in your CSV file, imported issues will be given the default (i.e. first) Resolution as specified in your Jira system <a href="#">Defining resolution field values</a> . You can also create new Jira values on-the-fly during the import process.  Also, see <a href="#">How to handle unresolved issues</a> for helpful tips.
Status	Can only be mapped to existing workflow statuses in Jira. If not specified in your CSV file, imported issues will be given the default (i.e. first) Status as specified in your Jira system.
Original Estimate	The value of this field needs to be specified as number of seconds.
Remaining Estimate	The value of this field needs to be specified as number of seconds.

Time Spent	The value of this field needs to be specified as number of seconds.
Users	<p>You can choose to have the importer automatically create Jira users for any values of the Assignee or Reporter field.</p> <ul style="list-style-type: none"> <li>• Users will be created as active accounts in Jira. Users will need to get their passwords emailed to them the first time they log into Jira.</li> <li>• Users with no real name will get the portion of their email address (login name) before the "@" character as their Full Name in Jira.</li> <li>• If you are using External User Management, the import process will not be able to create Jira users; instead, the importer will give you a list of any new users that need to be created. You will need to create the users in your external user repository before commencing the import.</li> <li>• If you have a user-limited license (e.g. personal license), and the number of required users is larger than the limit, then the import will be stopped. A page will be displayed showing a list of users that can't be created.</li> <li>• If Assignee and Reporter are not mapped, then no usernames are created</li> </ul>
Watchers	If you have users specified as Watchers in your CSV file, and these users do not exist in Jira, they will not be imported. A user must be available in Jira before you can import them as a watcher on a specific issue.
Other fields	If you wish to import any other fields, you can choose to map them to specific Jira custom field(s). If your custom fields don't yet exist in Jira, the importer can automatically create them for you. If your custom field is a date field, please use the date format specified on the second step of the CSV import wizard.

## Importing issues in bulk

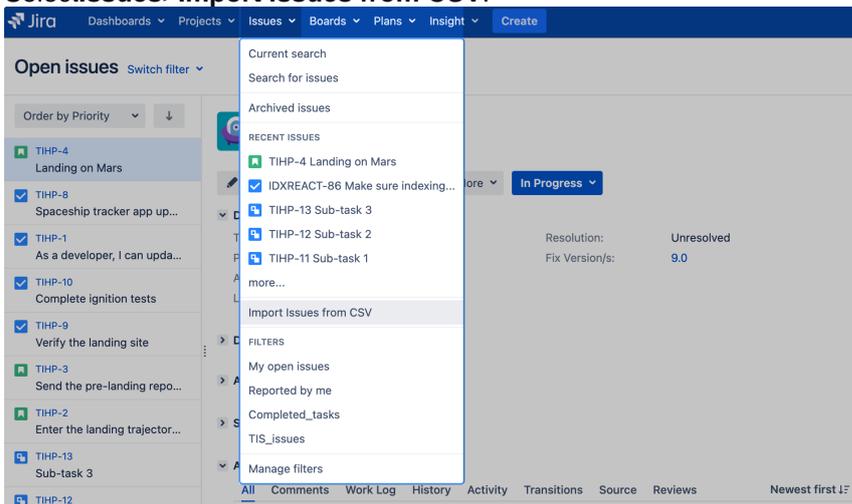
The Bulk issues import feature allows non-administrative users to import issues to Jira. External system import, which is described above, is available only for Jira Administrators.

With Bulk issues import, you should also import issues from a CSV file. But the entry point and steps to run Bulk issues import differ from External system import.

The Bulk issues import functionality has the following restrictions:

- New users won't be created.
- New projects won't be imported. You should import issues only to existing projects.
- Custom fields and resolution values can't be created in issues.
- Some fields in issues may be missing. See the complete list of such fields in [Missing fields for mapping when importing issues through CSV in Jira](#).

Prepare the CSV file with your Jira issues as described [here](#). To import the file:

1. Select **Issues** > **Import issues from CSV**.

2. Select the CSV file that you want to import.
3. Leave the **Use an existing configuration file** checkbox cleared if you don't have a configuration file or if you want to create a new configuration file. In this case, Jira will create the configuration file that you can use for further imports. If you select the checkbox, you should upload the existing configuration file.
4. Select the **Next** button.
5. Fill in the fields. Select the project where you want to import the issues, file encoding type, delimiter, and date format. If the CSV file uses a different separator, specify the character in the **Delimiter** field instead of a comma. If the separator is a tab, specify it as `/t`.
6. Select the **Next** button.
7. Map the fields in the CSV file to the issue fields in the selected project. If you want to map a specific field value in the CSV file to a specific Jira field value, select the **Map field value** checkbox.

**i** You must map at least one field in the CSV file to the Jira **Summary** field because each Jira issue must have a summary. If, instead of the Jira field for mapping, you see the **Don't map this field** note, it means that Jira can't provide the right mapping for the field from the CSV file. To solve this issue, see [Missing fields for mapping when importing issues through CSV in Jira](#).

8. Select the **Next** button.
9. If you selected the **Map field value** checkbox for some fields, you should map the values of these fields from the CSV file to the specific values of corresponding Jira issue fields. For example, you may want to map the CSV field value `Feature Request` to the Jira issue types field value `New Feature`.
10. If you want to check your configuration for errors or warnings before running the import, select **Validate**. If needed, you can download the detailed log of the validation.
11. If needed, save the configuration for further use. For example, you may want to use the same field or value mappings for the next imports.
12. Select **Begin Import** when you're sure of the configuration.

You've successfully bulk-imported your issues to Jira! If you have any questions or problems, contact [Atlassian support](#). If you're a Jira Administrator, we recommend using the External system import to avoid inconsistencies and errors.

# Commonly asked CSV questions and known issues

This page answers some of the commonly asked CSV questions our technical support staff have encountered. If you are not able to find an answer from this page and our [issue tracker](#), feel free to [create a support issue](#).

## Commonly Asked Questions

### The importer simply doesn't work on my CSV file!

Please make sure that it is a valid and not-bad-formatted CSV file. You should be able to spot this with by turning on detailed [logging and profiling](#). Also, please double check your configuration file and ensure that it's properly configured, e.g. exact delimiter, date format, etc.

### The importer fails at date fields, why?

If you are seeing error message similar to this:

```
[00:55:28] FAILED: Customfield value 01/Nov/06 12:00 AM is invalid
[00:55:28] com.atlassian.jira.issue.customfields.impl.FieldValidationException: Invalid date format. Please
enter the date in the format "MMM/dd/yy".
at com.atlassian.jira.issue.customfields.converters.DatePickerConverter.getTimestamp(DatePickerConverter.
java:57)
at com.atlassian.jira.issue.customfields.impl.DateCFType.getSingularObjectFromString(DateCFType.java:46)
at com.atlassian.jira.imports.importer.impl.DefaultJiraDataImporter.importIssues(DefaultJiraDataImporter.
java:531)
at com.atlassian.jira.imports.importer.impl.DefaultJiraDataImporter.doImport(DefaultJiraDataImporter.java:
104)
at com.atlassian.jira.imports.importer.impl.ImporterThread.run(ImporterThread.java:21)
```

There are a few possible reasons:

- The format of dates is not correctly set in the import configuration file. The date format for custom fields must match the "Date format in input file" which has a default format of `ofyyyyMMddHHmmss`
- Jira system date fields such as Created, Updated and Due Date use "yyyy-MM-dd HH:mm:ss" but may need an offset adding
- Date Picker and Date Time Picker formats are not consistent, e.g.

```
jira.date.picker.java.format=dd/MMM/yy
jira.date.time.picker.java.format=MMM/dd/yy hh:mm a
```

should be corrected to,

```
jira.date.picker.java.format=dd/MMM/yy
jira.date.time.picker.java.format=dd/MMM/yy hh:mm a
```

### Why does the importer always ask me to map values to column (at Step 3 of 5)?

It is because you have selected *Map Field Value* for the particular columns. To use the values from the CSV, you need just to map the column to the *Corresponding Jira field*, otherwise, select the *Map field value* checkbox.

## Known Issues

This is an open issue being tracked at [JRASERVER-45878](#).

This issue is being tracked at [JSWSERVER-16529](#).

There is a known problem that prevents the CSV Importer from being used with Jira instances running on JBoss 4.x. This is due to a compatibility issue between the JBoss 4.x commons-collections.jar and the Jira commons-collections.jar. The workaround is to replace the commons-collections.jar in JBoss 4.x with the more recent Jira version. Please see [JRA-6473](#) for further details.

# How to import CSV data with PVCS command

 The content on this page relates to platforms which are not supported for Jira. Consequently, Atlassian can not guarantee providing any support for it. Please be aware that this material is provided for your information only and using it is done so at your own risk.

Importing from PVCS is not supported yet, but there is a feature request being tracked [here](#). The above problem occurs when the pvcs command is not configured in the CSV configuration.

## Resolution

In order to import the author of the comment and the date of the comment successfully, there are a few required conditions:

- Append the settings in the csv configuration file which you have saved the configuration through the [wizard](#)

```
settings.advanced.mapper.comment : com.atlassian.jira.imports.csv.mappers.PvcsComment
```

-  For the latest plugin version 2.6.1, please use the configuration below:

```
settings.advanced.mapper.comment : com.atlassian.jira.plugins.importer.imports.csv.mappers.PvcsComment
```

- Username (Example: eddie) must exists in Jira
- The format of the comment should be as below:

```
"QA Note on Close: eddie: 4/28/2004 11:54:35 AM: Closing this defect as it is no longer relevant"
```

# Importing data from Excel

Unfortunately, right now we don't have a built-in Jira importer for native Microsoft Excel files. **But it is still possible to perform a two-stage import using CSV import mechanisms.**

## How to transform an MS Excel files into a CSV file

Microsoft Excel is capable of saving the spreadsheet in multiple file formats, including CSV. Before you save, we recommend you clean up the spreadsheet from all unnecessary information or macros and make sure that the table columns are labeled correctly.

When ready, select File / Save As and chose the CSV format from the "save as type" drop-down list.

In case of problems please refer to Microsoft documentation for help.

## How to import CSV data back to Jira

If you want to create issues as well as projects, users, etc. please refer to our [CSV importer help](#). If you don't have administrative privileges in Jira, you can also import CSV data into a single project through the user CSV importer, if enabled. In both cases, the importer wizards will guide you through the steps of mapping fields and values and validating the data before the import.

## If you are looking for an easier solution

The import through CSV has some limitations. If the results aren't satisfactory, there are complete third party solutions available which might help you. Please check out the following offers solution from Atlassian Marketplace:

- [Excel Connector for Jira](#) from Transition Technologies S.A.

You can also contact your local Atlassian Expert for help.

## Why do we have this page?

We are tracking the visits to this page. The intensity of visits will help us prioritize the work on the next set of Jira importers. It will not help you today, but think of yourself as of a democratic voter who will change the future of Jira. And for that we thank you.

# Importing data from Bitbucket



As of Jira 8.4, we no longer support built-in importers that are dedicated to specific applications, including this one. You can still import data to Jira in CSV or JSON format.

To import your data:

1. Log in to Jira as a user with the **Jira Administrators** [global permission](#).
2. Choose **Administration** (⚙️) > **System**. Select **Import & Export** > **External System Import** to open the Import external projects page.
3. Select CSV or JSON as the file format for your import.

# Importing data from Github



As of Jira 8.4, we no longer support built-in importers that are dedicated to specific applications, including this one. You can still import data to Jira in CSV or JSON format.

To import your data:

1. Log in to Jira as a user with the **Jira Administrators** [global permission](#).
2. Choose **Administration** (⚙️) > **System**. Select **Import & Export** > **External System Import** to open the Import external projects page.
3. Select CSV or JSON as the file format for your import.

# Importing Data from Asana



As of Jira 8.4, we no longer support built-in importers that are dedicated to specific applications, including this one. You can still import data to Jira in CSV or JSON format.

To import your data:

1. Log in to Jira as a user with the **Jira Administrators** [global permission](#).
2. Choose **Administration** (⚙️) > **System**. Select **Import & Export** > **External System Import** to open the Import external projects page.
3. Select CSV or JSON as the file format for your import.

# Importing data from TFS or Visual Studio

You can perform a two-stage import using Visual Studio's export mechanisms.

## How to export data from Visual Studio into a CSV file

This process has two steps.

In step one, you need to create a query with the work items that you want to export. When this a query is created, you can save its results into the Excel spreadsheet. (You might need to install Microsoft Excel add-in to Team Foundation Server first.)

In step two, you need to save the resulting spreadsheet into a CSV format.

Please refer to Microsoft Team Foundation Server and Visual Studio documentation for help.

## How to import CSV data back to Jira

If you want to create issues as well as projects, users, etc. please refer to our [CSV importer help](#). If you don't have administrative privileges in Jira, you can import CSV data directly into a single project with the user CSV importer. In both cases the importer wizards will guide you through the steps of mapping fields and values and validating the data before the import.

## If you are looking for an easier solution

The import through CSV has certain limitations. If the results aren't satisfactory, there are complete third party solutions available which might help you. Please check out the following solutions from Atlassian Marketplace:

- [TFS4Jira](#) from Spartez
- [UseTFS](#) from Pigsty
- [Jira Connector for ConnectALL](#) from Go2Group

You can also contact your local Atlassian Expert for help.

## Why do we have this page?

We are tracking the visits to this page. The intensity of visits will help us prioritize the work on the next set of Jira importers. It will not help you today, but think of yourself as of a democratic voter who will change the future of Jira. And for that we thank you.

# Importing data from Rally

You can perform a two-stage import using Rally's CSV export mechanisms.

## How to export data from Rally into a CSV file

It's possible to export data from Rally into CSV or XML files. However, CSV files are more reliable and we recommend them for the purpose of the migration.

In order to create a CSV file, go to the Rally's summary page and from the "actions" menu select the "CSV" option. Save the resulting file. This kind of export will only contain the data visible on the summary page. If you want to export all data you need to create a custom view first. Refer to Rally's documentation for help.

## How to import CSV data back to Jira

If you want to create issues as well as projects, users, etc. please refer to our [CSV importer help](#). If you don't have administrative privileges in Jira, you can also import CSV data into a single project through the user CSV importer, if enabled. In both cases, the importer wizards will guide you through the steps of mapping fields and values and validating the data before the import.

## If you are looking for an easier solution

The import through CSV has some limitations. If the results aren't satisfactory, there are complete third party solutions available which might help you. Please check out the following solutions from Atlassian Marketplace:

- [Rally to Jira Enterprise Migration Tool](#) from cPrimeLabs
- [Jira Connector for ConnectALL](#) from Go2Group
- [agosense.symphony](#) from agosense GMBH

You can also contact your local Atlassian Expert for help.

## Why do we have this page?

We are tracking the visits to this page. The intensity of visits will help us prioritize the work on the next set of Jira importers. It will not help you today, but think of yourself as of a democratic voter who will change the future of Jira. And for that we thank you.

# Importing data from VersionOne

You can perform a two-stage import using VersionOne's CSV export mechanisms.

## How to export data from VersionOne into a CSV file

In order to create a CSV file you need to use the VersionOne's custom reporting. Custom reporting allows you to perform an export to different file formats, including CSV. Refer to VersionOne's documentation for help on how to use custom reporting and exporting to CSV. Save the resulting file.

## How to import CSV data back to Jira

If you want to create issues as well as projects, users, etc. please refer to our [CSV importer help](#). If you don't have administrative privileges in Jira, you can also import CSV data into a single project through the user CSV importer, if enabled. In both cases, the importer wizards will guide you through the steps of mapping fields and values and validating the data before the import.

## If you are looking for an easier solution

The import through CSV has some limitations. If the results are not satisfactory, there are complete third party solutions available which might help you. Please check out the [Jira Connector for ConnectALL](#) from Go2Group on Atlassian Marketplace.

You can also contact your local Atlassian Expert for help.

## Why do we have this page?

We are tracking the visits to this page. The intensity of visits will help us prioritize the work on the next set of Jira importers. It will not help you today, but think of yourself as of a democratic voter who will change the future of Jira. And for that we thank you.

# Importing data from YouTrack

You can perform a two-stage import using VersionOne's CSV export mechanisms.

## How to export data from YouTrack into a CSV file

In order to create a CSV file you need to select the "Issues in CSV" option from your reports menu in YouTrack. Make sure to prepare the search criteria first so that exported data set is exactly what you want to have imported into Jira. Refer to YouTrack's documentation for help on how to use filters and reports. Save the resulting CSV file.

## How to import CSV data back to Jira

If you want to create issues as well as projects, users, etc. please refer to our [CSV importer help](#). If you don't have administrative privileges in Jira, you can also import CSV data into a single project through the user CSV importer, if enabled. In both cases the importer wizards will guide you through the steps mapping fields and values and validating the data before the import.

## If you are looking for an easier solution

The import through CSV has some limitations. If the results aren't satisfactory, please contact your local Atlassian Expert for help.

## Why do we have this page?

We are tracking the visits to this page. The intensity of visits will help us prioritize the work on the next set of Jira importers. It will not help you today, but think of yourself as of a democratic voter who will change the future of Jira. And for that we thank you.

# Importing data from Axosoft

You can perform a two-stage import using Axosoft's CSV export mechanisms.

## How to export data from Axosoft into a CSV file

In order to create a CSV file you need to go to your list of items or work logs and select the "Export" option from the "More" menu. Make sure to select all fields for the export, otherwise some information may not be visible in Jira. Save the resulting CSV file.

## How to import CSV data back to Jira

If you want to create issues, projects, users, etc, please refer to our [CSV importer help](#). If you don't have administrative privileges in Jira, you can also import CSV data into a single project through the user CSV importer, if enabled. In both cases, the importer wizards will guide you through the steps of mapping fields and values and validating the data before the import.

## If you are looking for an easier solution

The import through CSV has some limitations. If the results aren't satisfactory, please contact your local Atlassian Expert for help.

## Why do we have this page?

We are tracking the visits to this page. The intensity of visits will help us prioritize the work on the next set of Jira importers. It will not help you today, but think of yourself as of a democratic voter who will change the future of Jira. And for that we thank you.

# Importing data from Pivotal Tracker



As of Jira 8.4, we no longer support built-in importers that are dedicated to specific applications, including this one. You can still import data to Jira in CSV or JSON format.

To import your data:

1. Log in to Jira as a user with the **Jira Administrators** [global permission](#).
2. Choose **Administration** (⚙️) > **System**. Select **Import & Export** > **External System Import** to open the Import external projects page.
3. Select CSV or JSON as the file format for your import.

# Importing data from Bugzilla



As of Jira 8.4, we no longer support built-in importers that are dedicated to specific applications, including this one. You can still import data to Jira in CSV or JSON format.

To import your data:

1. Log in to Jira as a user with the **Jira Administrators** [global permission](#).
2. Choose **Administration** (⚙️) > **System**. Select **Import & Export** > **External System Import** to open the Import external projects page.
3. Select CSV or JSON as the file format for your import.

# Importing data from FogBugz On Demand

 As of Jira 8.4, we no longer support built-in importers that are dedicated to specific applications, including this one. You can still import data to Jira in CSV or JSON format.

To import your data:

1. Log in to Jira as a user with the **Jira Administrators** [global permission](#).
2. Choose **Administration** () > **System**. Select **Import & Export** > **External System Import** to open the Import external projects page.
3. Select CSV or JSON as the file format for your import.

# Importing data from FogBugz for your Server



As of Jira 8.4, we no longer support built-in importers that are dedicated to specific applications, including this one. You can still import data to Jira in CSV or JSON format.

To import your data:

1. Log in to Jira as a user with the **Jira Administrators** [global permission](#).
2. Choose **Administration** (⚙️) > **System**. Select **Import & Export** > **External System Import** to open the Import external projects page.
3. Select CSV or JSON as the file format for your import.

# Importing data from Trac



As of Jira 8.4, we no longer support built-in importers that are dedicated to specific applications, including this one. You can still import data to Jira in CSV or JSON format.

To import your data:

1. Log in to Jira as a user with the **Jira Administrators** [global permission](#).
2. Choose **Administration** (⚙️) > **System**. Select **Import & Export** > **External System Import** to open the Import external projects page.
3. Select CSV or JSON as the file format for your import.

# Importing data from Redmine



As of Jira 8.4, we no longer support built-in importers that are dedicated to specific applications, including this one. You can still import data to Jira in CSV or JSON format.

To import your data:

1. Log in to Jira as a user with the **Jira Administrators** [global permission](#).
2. Choose **Administration** (⚙️) > **System**. Select **Import & Export** > **External System Import** to open the Import external projects page.
3. Select CSV or JSON as the file format for your import.

# Importing data from BaseCamp

Unfortunately, it's not possible to export the data from Basecamp into a file format which can be directly imported back to Jira.

There are third party solutions available which might help you. Please check out the [TaskAdapter](#) solution from Atlassian Marketplace. You can also contact your local Atlassian Expert for help or [develop](#) a solution based on Basecamp and [Jira public APIs](#).

# Importing data from JSON

Version 4.3 or later of the [Jira Importers plugin](#), which is bundled with Jira, allows you to import data from a JavaScript Object Notation (JSON) file.

JSON files are easy to read and encapsulate more structure and information than CSV files.

The JSON import feature allows you to import issues from an external (issue tracking) system which can export its data in a JSON format.

You may also wish to prepare your JSON file manually.

**⚠** Please note that the import format used by the Jira Importers plugin is more basic than the import format available when using the Jira REST API.

## Creating a JSON file for Import

If your current issue tracking system is unable to export in the JSON format, you may wish to create the file manually. To prepare the JSON file, you should use the standard [JSON format](#), and follow the pattern detailed below.

### On this page:

- [Creating a JSON file for Import](#)
- [Running the JSON File Import Wizard](#)
- [Known issues:](#)

#### JSON File Example

```
{
  "users": [
    {
      "name": "alice",
      "fullname": "Alice Foo"
    },
    {
      "name": "bob",
      "fullname": "Bob Bar"
    }
  ],
  "links": [
    {
      "name": "sub-task-link",
      "sourceId": "2",
      "destinationId": "1"
    },
    {
      "name": "Duplicate",
      "sourceId": "3",
      "destinationId": "2"
    }
  ],
  "projects": [
    {
      "name": "A Sample Project",
      "key": "ASM",
      "type": "software",
      "description": "JSON file description",
      "versions": [
        {
          "name": "1.0",
          "released": true,
          "releaseDate": "2012-08-31T15:59:02.161+0100"
        },
        {
          "name": "2.0"
        }
      ]
    }
  ],
  "components": [
    "Component",
    "AnotherComponent"
  ],
  "issues": [
    {
      "priority": "Major",
      "description": "Some nice description here\nMaybe italics or bold?",
    }
  ]
}
```

```

        "status" : "Closed",
        "reporter" : "alice",
        "labels" : [ "impossible", "to", "test" ],
        "watchers" : [ "bob" ],
        "issueType" : "Bug",
        "resolution" : "Resolved",
        "created" : "2012-08-31T17:59:02.161+0100",
        "updated" : "P-1D",
        "affectedVersions" : [ "1.0" ],
        "summary" : "My chore for today",
        "assignee" : "bob",
        "fixedVersions" : [ "1.0", "2.0" ],
        "components" : [ "Component", "AnotherComponent" ],
        "externalId" : "1",
        "history" : [
          {
            "author" : "alice",
            "created": "2012-08-31T15:59:02.161+0100",
            "items": [
              {
                "fieldType" : "jira",
                "field" : "status",
                "from" : "1",
                "fromString" : "Open",
                "to" : "5",
                "toString" : "Resolved"
              }
            ]
          }
        ],
        "customFieldValues": [
          {
            "fieldName": "Story Points",
            "fieldType": "com.atlassian.jira.plugin.system.customfieldtypes:float",
            "value": "15"
          },
          {
            "fieldName": "Business Value",
            "fieldType": "com.atlassian.jira.plugin.system.customfieldtypes:float",
            "value": "34"
          }
        ],
        "attachments" : [
          {
            "name" : "battarang.jpg",
            "attacher" : "admin",
            "created" : "2012-08-31T17:59:02.161+0100",
            "uri" : "http://optimus-prime/~batman/images/battarang.jpg",
            "description" : "This is optimus prime"
          }
        ]
      },
      {
        "status" : "Open",
        "reporter" : "bob",
        "issueType": "Sub-task",
        "created" : "P-3D",
        "updated" : "P-1D",
        "summary" : "Sub-task",
        "externalId": "2"
      },
      {
        "status" : "Closed",
        "reporter" : "alice",
        "issueType": "Sub-task",
        "created" : "P-3D",
        "updated" : "P-1D",
        "resolution" : "Duplicate",
        "summary" : "Duplicate Sub-task",
        "externalId": "3"
      }
    ]
  }
}

```

## Custom Fields

The JSON Importers plugin supports custom fields. Below is a list of custom fields that come bundled with Jira. If you have installed any additional plugins that have custom fields, these fields will also be supported, however they are not included in this list.

1. [com.atlassian.jira.plugin.system.customfieldtypes:textfield](#)
2. [com.atlassian.jira.plugin.system.customfieldtypes:textarea](#)
3. [com.atlassian.jira.plugin.system.customfieldtypes:datepicker](#)
4. [com.atlassian.jira.plugin.system.customfieldtypes:datetime](#)
5. [com.atlassian.jira.plugin.system.customfieldtypes:float](#)
6. [com.atlassian.jira.plugin.system.customfieldtypes:select](#)
7. [com.atlassian.jira.plugin.system.customfieldtypes:radiobuttons](#)
8. [com.atlassian.jira.plugin.system.customfieldtypes:project](#)
9. [com.atlassian.jira.plugin.system.customfieldtypes:multiversion](#)
10. [com.atlassian.jira.plugin.system.customfieldtypes:version](#)
11. [com.atlassian.jira.plugin.system.customfieldtypes:userpicker](#)
12. [com.atlassian.jira.plugin.system.customfieldtypes:url](#)
13. [com.atlassian.jira.plugin.system.customfieldtypes:multiselect](#)
14. [com.atlassian.jira.plugin.system.customfieldtypes:multicheckboxes](#)
15. [com.atlassian.jira.plugin.system.customfieldtypes:multiuserpicker](#)
16. [com.atlassian.jira.plugin.system.customfieldtypes:multigrouppicker](#)
17. [com.atlassian.jira.plugin.system.customfieldtypes:grouppicker](#)
18. [com.atlassian.jira.plugin.system.customfieldtypes:cascadingselect](#)
19. [com.atlassian.jira.plugin.system.customfieldtypes:readonlyfield](#)
20. [com.atlassian.jira.plugin.system.customfieldtypes:labels](#)

The custom field example below shows some syntax for adding custom fields, including an example of a cascading custom field. If the custom field is not listed above, the "fieldType" can be obtained from the Custom Fields configuration page, by inspecting the source HTML. The "value" is specific to each custom field, and you can find this by inspecting the Edit Issue page's source HTML.

Custom Field Example	
<pre> "customFieldValues": [   //Custom Fields which accepts single values:   {     "fieldName": "My Awesome Text Field (single line)",     "fieldType": "com.atlassian.jira.plugin.system.customfieldtypes:textfield",     "value": "some text"   },   {     "fieldName": "My Awesome Select List (single choice)",     "fieldType": "com.atlassian.jira.plugin.system.customfieldtypes:select",     "value": "some select"   },   //Custom Fields which accepts multiple values:   {     "fieldName": "My Awesome Checkboxes",     "fieldType": "com.atlassian.jira.plugin.system.customfieldtypes: multicheckboxes",     "value": [ "multiple", "checkboxes" ]   },   {     "fieldName": "My Awesome User Picker (multiple users)",     "fieldType": "com.atlassian.jira.plugin.system.customfieldtypes: multiuserpicker",     "value": [ "admin", "fred" ]   },   //Custom Fields which accepts Options in hierarchy. That's only cascading select   from standard JIRA pool.   {     "fieldName": "My Awesome Select List (cascading)",     "fieldType": "com.atlassian.jira.plugin.system.customfieldtypes: cascadingselect",     "value":     {       "": "Parent Value",       "1": "Child Value"     }   } ] </pre>	

## Specific JSON File Examples

Further specific JSON file examples include:

Supported Field	Notes	Example
Users	This example covers a full user. In this example, two groups have been specified. If a group does not exist already, the Jira Importers plugin will create it.	<div style="border: 1px solid #ccc; padding: 5px;"> <p><b>User Example</b></p> <pre> "users": [   {     "name" : "someuser",     "groups" : [ "jira-users", "my- custom-group" ],     "active" : true,     "email" : "user1@example.com",     "fullname" : "User 1"   } ] </pre> </div>

Version	To import an issue and specify, for example, a fixVersion, this fixVersion needs to be defined in the JSON file. Even if the Version is already in JIRA it must also be specified in JSON under projects.	<p><b>Version</b></p> <pre> "versions": [   {     "name": "version_1"   } ], </pre>
Project Key and Issue Key	You can assign a key to both the project and the issue. These keys can be different. This example will create a project with one issue, "SAM-123".	<p><b>Project Key and Issue Key Example</b></p> <pre> {   "projects": [     {       "name": "Sample data",       "key": "SAM",       "type": "software",       "issues": [         { "key" : "SAM-123",           "status" : "Open",           "reporter" : "admin",           "summary" : "Parent case",           "externalId": "123"         }       ]     }   ] } </pre>

<p>Comments</p>	<p>This example shows how you can import multiple comments for an issue.</p>	<p><b>Comment Example</b></p> <pre> {   "projects": [     {       "name": "Sample data",       "key": "SAM",       "issues": [         {           "status": "Open",           "reporter": "admin",           "summary": "Parent case",           "externalId": "1",           "comments": [             {               "body": "This is a comment from admin 5 days ago",               "author": "admin",               "created": "2012-08-31T17:59:02.161+0100"             },             {               "body": "This is a comment from admin 1 day ago",               "author": "admin",               "created": "2012-08-31T17:59:02.161+0100"             }           ]         }       ]     }   ] } </pre>
<p>Worklogs</p>	<p>This example shows the syntax to import worklog detail.</p>	<p><b>Worklog Example</b></p> <pre> "worklogs": [   {     "author": "admin",     "comment": "Worklog",     "startDate": "2012-08-31T17:59:02.161+0100",     "timeSpent": "PT1M"   },   {     "author": "admin",     "startDate": "2012-08-31T17:59:02.161+0100",     "timeSpent": "PT3H"   } ] </pre>

<p>Component</p>	<p>Components can be specified in a JSON file in two ways, by providing a name, or by providing an object. This example shows both. The Jira Importers plugin will always create a new component with "Default Assignee" switched to "Project Default", as you are unable to specify a "Default Assignee".</p>	<p><b>Component Example</b></p> <pre> "components": [   "Component",   //Component specified only by name   { // Component     specified by object     "name": "SomeName",     "lead": "admin",     "description":     "Some description"   } ], </pre>
<p>Issues with Time Tracking</p>	<p>Time Tracking detail can be imported with an issue. This example shows you an issue with Time Tracking detail. The "originalEstimate", "timeSpent", and "estimate" values must be in Period format (<a href="#">Format ISO_8601 - Durations</a>). The "startDate" value accepts both the DateTime and Period format.</p> <p>Please ensure Time Tracking is enabled in Jira before you start your import, otherwise the data will be ignored by the Jira Importers plugin during the import.</p>	<p><b>Issues with Time Tracking</b></p> <pre> "issues": [   {     "summary": "My Example Time Tracking issue",     "externalId": "1",     "originalEstimate":     "P1W3D",     "timeSpent": "PT4H",     "estimate": "P2D",     "worklogs": [       {         "author":         "admin",         "comment":         "Worklog",         "startDate": "P-1D", //can be a Period or DateTime         "timeSpent": "PT1M"       },       {         "author":         "admin",         "startDate": "2014-01-14T17:00:00.000 +0100",         "timeSpent": "PT3H"       }     ]   } ] </pre>

**i** Dates can be represented in [SimpleDateFormat](#) "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (example output: "2012-08-31T15:59:02.161+0100") or you can use relative dates like "P-1D" (which means one day ago).

## Running the JSON File Import Wizard

Before you begin, please [back up](#) your Jira data.

1. Log in to Jira as a user with the [Jira Administrators global permission](#).
2. Choose **Administration**  > **System**. Select **Import & Export > External System Import** to open the Import external projects page.
3. Select **JSON** to open the **JSON File import** page.

4. Choose your JSON file.
5. Click the **Begin Import** button when you are ready to begin importing your JSON file into Jira. The importer will display updates as the import progresses, then a success message when the import is complete.

**Note:** If you experience problems with the import (or you are just curious), click the **download a detailed log** link to view detailed information about the JSON file import process. This information can also be useful if you encounter any errors with your import.

Congratulations! You have successfully imported your JSON projects into Jira! If you have any questions or encounter any errors, please contact [Atlassian support](#).

#### Known issues:

It's been reported that when attachments contain characters issues get created without these attachments. See [JRASERVER-64674](#).

# Importing data from Mantis



As of Jira 8.4, we no longer support built-in importers that are dedicated to specific applications, including this one. You can still import data to Jira in CSV or JSON format.

To import your data:

1. Log in to Jira as a user with the **Jira Administrators** [global permission](#).
2. Choose **Administration** (⚙️) > **System**. Select **Import & Export** > **External System Import** to open the Import external projects page.
3. Select CSV or JSON as the file format for your import.

# Moving or archiving individual projects

Over time, your organization's requirements may change. You may need to:

- [Archive](#) a completed or obsolete project.
- [Split](#) a large Jira instance into several Jira instances, with particular projects in each.
- [Restore](#) a single project from a backup file into a Jira instance.
- [Restore](#) an entire Jira instance, from a backup into a new empty Jira instance.

# Splitting Jira applications

Occasionally, an organization may need to split its existing Jira application instance into two separate instances. For example, there might be a requirement to have some particular projects in one instance, and other projects in a second instance.

## Note

This process requires two separate server licenses.

1. Back up your [database](#), using your database backup procedures, and verify the backup.
2. Back up your attachments directory and verify the backup.
3. Install the needed Jira applications (e.g. Jira Software) on your new server.
  - ⚠ **Please Note:**
    - The Jira application version number on your new server must be the same as (or higher than) the version number on your existing server.
    - Do not use the same [Jira home directory](#) for the two Jira application instances. Specify a new Jira application home directory for the Jira application on your new server.
    - Do not connect the two Jira application instances to the same external database instance.
4. Create an XML backup from your existing Jira server application, as described in [Backing up data](#).
5. Import the XML backup file into your new server, as described in [Restoring data](#).
6. Copy the attachments directory from your existing server to your new server, and configure your new server to use its own directory. See [Configuring file attachments](#) for more information.
7. At this point you should have two Jira application instances with the same users, projects, issues and attachments. Log in to both instances and perform some random searches to verify that the data is identical in both instances.
8. Delete the non-required projects from each Jira application.
9. Generate new Server ID for the newly installed Jira application, as described in the article [Changing Server ID](#). This step is needed if you plan to create [Application Links](#) between the two instances.

# Exporting issues from Cloud to Server

If you already have a Jira Cloud site and want to move to Jira Server, you can create a backup of your Jira Cloud data that you can then import into a server installation. Note that the Atlassian Cloud service takes backups for your instance every 24 hours for purposes of application recovery (not for rolling back application data).

You can backup and export the following data from your Jira Cloud site:

- Issues
- Users and user group settings
- Issue attachments, user avatars, and project logos (if selected)

## On this page:

- [How to create a backup](#)
- [How to structure the export file](#)
- [How to import backup data into Jira Cloud](#)
- [How to import backup data into Jira Server](#)

## How to create a backup

You can generate a new backup every 24 hours from the time the previous backup has finished. Note that Jira only stores one backup file at a time, and any existing existing backup will be overwritten by a new one. To generate a backup:

1. In the upper-right corner of the screen, select **Administration**  > **System**.
2. In the Import and Export section, click **Backup manager**.
3. Check the additional files option if you want to include issue attachments, user avatars, and project logos in the export.
4. Select the type of backup:
  - If you will restore the data to a cloud instance, select **Create Backup for Cloud**
  - If you will to restore the data to a server instance, select **Create Backup for Server**

After the backup is complete, click the file link to download the backup.

In some cloud sites where the Backup Manager menu does not appear, you can use one of the following workarounds to access it:

`https://<domain_name>.atlassian.net/plugins/servlet/ondemandbackupmanager/admin`

`https://<account_name>.jira.com/plugins/servlet/ondemandbackupmanager/admin`

## How to structure the export file

Once you have generated and unzipped a backup file, you should have an output similar to the following:

```
Jira-backup-20161021
activeobjects.xml
entities.xml
data
  attachments
  avatars
logos
```

Note that database data is stored in both `activeobjects.xml` and `entities.xml`. Issue attachments, user avatars, and project logos are stored in corresponding directories. If you don't want to import attachments, avatars, or logos to your new Jira Cloud site, you can remove the corresponding directory and then zip the modified directory tree before importing.

## How to import backup data into Jira Cloud

1. Structure your export file as mentioned [above](#).
2. Follow the instructions on [importing issues](#).
3. After the import, log into your new Jira Cloud site with the same admin account you used to log into your original site. Cross-application links (links to a source file page from a Jira issue) will point back to the corresponding source location after the import.

## How to import backup data into Jira Server

1. Structure your export file as mentioned [above](#).
2. Follow the instructions on the [Migrating from Jira Cloud to Jira Server](#) page.
3. After the import, log into your Jira Server instance with the `sysadmin` username and corresponding password. Make sure to change your password after you log in. Cross-application links (links to a source file page from a Jira issue) will point back to the corresponding source location after the import.

 Note that the `sysadmin` user is created automatically during the backup process. This user is created to allow you to log in with the necessary Jira System Administrator permission (a server-specific permission not available in the cloud) after restoring data in Jira.

## Exporting issues from Server to Cloud

When opened in a viewport, the user will be redirected to: [Jira server to cloud migration resources](#).

# Migrating data with 3rd party apps

Whether you're scaling your organization, simplifying maintenance for hardware and licensing, or taking an extra step to validate all changes coming into Jira, we want to ensure there's a clear path for you to migrate data. That's why we teamed up with two of our Top Vendors, **Adaptavist** and **Botron**, who each offer an app to help you transfer projects, configuration, and the desired accompanied data from one Jira instance to another. Why two vendors? While both ensure the same goal, they each offer a different approach to get there. Read more to choose the one that works best for your team.



### Promoting changes in Jira

Whether it's a big change to your workflows or just a new custom field, take it from development through staging to production to make sure it's all checked up.

[Adaptavist: Promoting Jira configuration](#)

[Botron: Promoting Jira configuration](#)

---



### Consolidating Jira instances

You can never get too much of Jira, but too many instances might sometimes feel overwhelming. When moving to a single Jira, don't leave your data behind.

[Adaptavist: Consolidating Jira instances](#)

[Botron: Consolidating Jira instances](#)

---



### Migrating Jira projects

If your projects are living out of a suitcase, you can at least make sure it's the one with wheels. Move them freely between one Jira and another.

[Adaptavist: Migrating Jira projects](#)

[Botron: Migrating Jira projects](#)

# Migrating data with Adaptavist

An add-on for Jira prepared by Adaptavist helps you transfer your configuration, projects, and all associated data from one Jira to another. Read about possible scenarios and choose the one you need.

## Promoting Jira configuration

[Learn more](#) about moving your changes from development through staging to production to make sure they're all checked up. By doing so, you protect your production environment from any mistakes or unevaluated changes.

## Consolidating Jira instances

[Learn more](#) about consolidating multiple Jira instances into a single one.

## Migrating Jira projects

[Learn more](#) about migrating your projects, along with all relevant data, from one Jira instance to another.

# Promoting configuration changes from staging to production

 Tasks described on this page are not native to Jira. To complete them, you'll need to install a 3rd party add-on [Project Configurator for Jira](#).

## Overview

- [Overview](#)
- [Preparing](#)
- [Developing and validating changes in staging](#)
- [Exporting configuration for changed projects from staging](#)
- [Importing project configuration into production](#)
- [Verifying results of promotion](#)
- [Troubleshooting](#)

## What do we mean by "promoting configuration changes from staging to production"?

Let us explain with an example. Imagine you are the Jira administrator in charge of a Jira instance with a few hundred users. The users working in two of the biggest projects want to implement changes to how those projects are managed in Jira. The Jira team knows that implementing those changes needs at least three days of work, and also that they should be reviewed and validated by the users before putting them in production. Some of the user requirements are not absolutely clear and you know that until users see the implementation they will not be able to decide if that new implementation is what they actually need. A preliminary analysis of the requirements has shown that they imply changes to project workflows, screens, custom fields, and permission schemes. New groups will also be created so that permissions can be granted only to those users that need them.

Starting to implement these changes on production means Jira users would have to put up with three days of partially implemented changes. What's more, the changes won't be final until the users validate them, so it is possible they have to be modified and redone a few times. This would mean massive disruption to everyday work for users in the affected projects. So you quickly realize that those changes must be implemented first in a separate development instance (which we will call "staging" in this guide.)

This decision brings up new questions:

- How to migrate the changes implemented in staging to production without manually redoing them?
- How to make sure the tests and validations in staging really represent the final result when the changes are moved into production?

This guide intends to answer these questions, offering a recommended process for implementing, reviewing, and promoting configuration changes between Jira instances. This process is based on the use of the plugin called [Project Configurator for Jira](#).

### A more elaborate process

In some cases, when the rate of changes applied into production is very high with frequent configuration updates for different projects, the process described in this guide might not guarantee a representative test of the new configuration impact on production. In these cases, it would be practical to have a variation of the process that uses an additional Jira instance.

In this variation, when changes are ready to be applied to production, a clone of production is created. Let's call this clone "pre-production". Changes are imported into pre-production and the new configuration is validated there, eventually fixing anything if necessary. Then, the new configuration for the changed projects is exported from pre-production and imported into production.

Obviously, the import into pre-production does not require the same caution as into a real production instance, so a previous backup and locking up the instance would not be needed.

## What is included in the concept of "configuration"?

The best way to explain it is pointing to the [list of entities that Project Configurator for Jira will move in a configuration export/import](#).

When you export the configuration for a group of projects, all configuration objects that are used by those projects will be exported. For more information, see [Selection of objects to export](#).

## Limitations

Technical limitations are described [here](#). Note that the limitations in the [items that must coincide in both Jira instances](#) section would not be an issue if the advice in this guide is followed. Starting with the staging instance as a clone of production would guarantee that those aspects of both instances are the same.

## Cloud and Server instances

The process explained in this guide is valid only for Jira Server instances. If you want to use it for promoting changes into a Jira Cloud instance, you could only do it *indirectly* by following these steps:

1. [Clone a Jira Cloud instance into a Server instance](#) that will be used as a staging machine (for development and test.) This would replace the [first step](#) explained in the guide.
2. Follow the rest of steps as explained in the guide, up to and including [exporting the configuration changes from staging](#).
3. Lock the Jira Cloud instance to users.
4. Clone it to a Jira Server instance. This will act as the "production" instance in the next steps of the process: [importing configuration](#) and [verifying results](#).
5. When verification shows that promotion results are correct at the "production" Server instance, [move it to the Cloud](#), replacing the previous Cloud instance.
6. Open the new Cloud instance to users and resume normal work.

# Preparing

## Get and install Project Configurator for Jira

Project Configurator for Jira is [available at the Atlassian Marketplace](#) as a Paid-via-Atlassian Plugin 2 add-on. It can be installed and licensed in all [the usual ways](#), either from the Universal Plugin Manager inside Jira, or from [its page](#) at the Marketplace.

The plugin must be available in all instances where it will be used for exporting or importing.

## Start a staging instance as a clone of production

You should start with a staging instance that is a copy of your production machine. This will ensure that configuration changes applied in staging cause the same effects they will have when applied to production. In other words, this will make tests and validation on staging representative of what will happen later when the configuration changes are moved into production.

For more information on how to clone a Jira instance, see [Establishing staging server environments](#).

## Synchronize staging with production

If the staging instance was cloned from production some time ago, it is a good idea to "refresh it" with the latest configuration changes from production. Remember as the staging instance is a closer reflection of production, the tests performed in staging will be more representative of the effects of the new configuration when it is finally promoted to production.

If the Jira Install directory at production hasn't been changed (for example, installing new add-ons or upgrading existing ones) since it was last cloned to staging, you don't need to repeat the whole cloning process. Just make a backup of production and restore it at staging. See [Backing up data](#) and [Restoring data](#).

## Agree with users on promotion windows

It is a good idea to agree previously with users when they will be available to review and validate changes at the staging instance.

As a safety measure, we will take a backup of the production Jira before promoting configuration changes to it. This implies that the contents of that instance must be frozen after the backup so that, if the need arises to restore the backup, there won't be any last-minute changes that are lost. So, it will be very convenient to find in advance candidate time slots where the following operations can be performed on production:

- Locking the instance
- Creating a backup
- Promoting configuration changes from staging
- Validating these changes
- Unlocking the instance

Very likely, qualified users will have to be a part of this decision.

Let us know what you think

[Feedback](#)

# Developing and validating changes in staging

## Change the configuration in staging

Now, it's time to change the required items on staging custom fields, workflows, schemes, etc. in order to satisfy the requirements that have been identified.

## Validate those changes internally, analyze impact on other projects

Review and validate the new configuration. Depending on the complexity and scope of proposed changes, it may be important to check if:

- The workflows are complete, with all required states and transitions.
- Issues have fields to hold all required information.
- Every issue has all its required fields filled, either at creation or at some point in its workflow.
- Everybody is granted permissions in line with tasks and responsibilities, but not more.
- All screens are verified, as they will be seen by end users.
- All main use cases are verified create, transition, and edit typical issues, use filters, dashboards, or reports to obtain the required information.

It is especially important that attention is paid to the impact of these changes on other projects. This impact is produced when you modify a configuration object (e.g. a workflow scheme) that is shared with several projects. Most often, Jira will show you where a configuration object is used. Following the example, the next image displays the workflow schemes administration page, where you can see which projects use every active workflow scheme:

The screenshot shows the Jira Administration interface for 'Workflow schemes'. The page title is 'Administration' with a search bar and a 'Back to project: Project Configurator Plugin' button. The navigation menu includes 'Applications', 'Projects', 'Issues', 'Add-ons', 'System', 'User management', 'Billing', and 'Discover new applications'. The left sidebar lists 'ISSUE TYPES', 'WORKFLOWS', and 'SCREENS'. The main content area is titled 'Workflow schemes' and includes an 'Add workflow scheme' button. Below the title, it states 'Workflow Schemes allow you to define which workflows apply to given issue types and projects.' There are two sections: 'Active' and 'Inactive'. The 'Active' section contains a table with the following data:

Name	Projects	Issue Type	Workflow	Actions
classic	• Project Configurator Plugin	Unassigned Types	classic default workflow	Edit Copy

The 'Inactive' section states 'There are no inactive workflow schemes.'

If you detect that a configuration object is being changed and it is used by other projects, you should analyze the impact of the changes on those other projects, eventually performing some of the reviews and tests mentioned above. The most complex situation would be that you detect that the proposed changes are incompatible with other projects when the impact on them is not acceptable. In this case, your best option is to copy the configuration object that is to be modified, so that one of the copies can be changed, while the other (used by other projects) remains untouched.

Note: Changes to custom fields with impacts on other projects can be ignored if the **Smart custom field contexts** option is enabled during the promotion of changes to production.

## Validate changes with users

When all changes are implemented in staging they can be shown to the users and reviewed jointly with them. The goal is getting the users to view and approve the new configuration, or gather feedback about desired changes.

Note that in some cases this user acceptance will be mandatory, according to the organization procedures.

## Knowing what has changed

Perhaps at some point during the development, you may be asking yourself what has changed in the configurations since you started from the production clone. There are two ways to get a practical answer to this question:

1. [Export the configuration](#) from the staging instance, and [run a simulated import](#) into production. This will print all differences between the configuration for the exported projects in staging and their *current* configuration in production.
2. Before starting developments in staging, [export the existing configuration](#) for the projects and save the resulting XML configuration file. When you want to check what has changed as a result of the new developments, export the new configuration into another XML file and compare it to the one you obtained before the development started. Any diff tool for text files available in your environment can be used for the comparison and the output XML file is designed so that it can be used to track configuration changes. This method has the advantage that you can store XML files of different stages of the development and then use them as a "version control" of the whole stream of configuration changes that are being made in the project. You could compare and view configuration changes in different moments in time and even restore the configuration in the staging instance to one of those past configurations ([importing the configuration](#) of that XML file into staging.)

Let us know what you think

[Feedback](#)

# Exporting configuration for changed projects from staging

When all the configuration changes are ready in staging, you can promote them to production. The first step is exporting them from staging.

## Starting the export

1. In the staging instance, open a Jira session as a user with the system administrator permission.
2. In the top-right corner, open the Administration menu, and click **Add-ons**.
3. In the menu on the left, in the PROJECT CONFIGURATOR section, click **Export selected projects**.

### ISSUE COLLECTORS

[Issue Collectors](#)

### PROJECT CONFIGURATOR

[Import project configuration](#)

[Import complete project](#)

[Export all projects](#)

[Export selected projects](#)

### MONITORING

4. Select the projects that you want to export (to select more than one, use the CTRL key in Windows, or COMMAND key in macOS.)
5. Select **Configuration only** below the list of projects, and click **Next**.

The screenshot shows the Jira Administration interface. At the top, there is a search bar for "Search JIRA admin" and a navigation menu with "Applications", "Projects", "Issues", "Add-ons", "User management", and "System". The "Add-ons" tab is active. On the left sidebar, under "PROJECT CONFIGURATOR", the "Export selected projects" option is highlighted. The main content area is titled "Project export" and features a progress bar with three steps: "Select projects", "Export options", and "See results". The "Select projects" step is currently active. Below the progress bar, there is a list of projects to export: "Temporary FIRST", "Temporary PA", "Temporary PB", "Temporary PC", "Temporary SECOND", and "Temporary THIRD". The "Temporary PA" and "Temporary PB" items are selected, indicated by a blue highlight. Below the list, there are radio buttons for "Export:" with options "Configuration only" (selected) and "Configuration and data". A "Next" button is located at the bottom of the form.

## Export options

On the next page, you can fine tune some export options. The meaning of each option is explained on [this page](#). Let's review them and see which are usually the best choice.

- **Filtering custom fields** In most cases, you will export only the custom fields that are used by the exported projects. Exporting all custom fields would add unnecessary complexity to the process. This option is kept mostly for historical reasons and backward compatibility.
- **User export options and Group export options** The choice here depends to a great extent on the management model in place for users and groups, and their part in the changes that have been performed in staging.
  - Full export Choose if users and groups are managed in the Jira internal directory, and have been created or changed in staging.
  - Do not export Choose if users and groups are managed in an external directory (Active Directory, any other kind of LDAP, etc.), or have *not* been changed in staging.
  - Do not export and Ignore invalid users/groups These might be useful when you have a large number of users and groups, and some of them might be inconsistent, e.g. there are references to users and groups that do not exist any longer, or have invalid email addresses.

Bear also in mind that if the production instance has several directories for users and groups, any new user /group will be created in the first writable directory (see [here](#) for more details.)

### Exporting filters, dashboards, and Agile boards

If the changes developed in staging are centered on a small group of projects, you can export only filters, dashboards, and Agile boards that are related to these projects. In such a case:

- For filters and dashboards, select the **Shared with exported projects** option.
- For Scrum and Kanban boards, select the **Associated to exported projects** option.

The criteria to decide when one of these objects is shared or associated with a project are these:

**Filters and dashboards** They are shared with the project or any of its roles.

**Scrum and Kanban boards** The board appears under the project title at the project navigation menu. This happens when the main filter for the board includes a clause like "project=XXX" that restricts its issues to that project.

### Launching the export

Finally, click **Export project configuration**, and you will navigate to a page with a progress bar that represents the progress of the export task. When the export is completed, your browser will download an XML file with the configuration for the selected projects. Store the downloaded XML file at a known location.

Let us know what you think

[Feedback](#)

# Importing project configuration into production

The next step in the promotion process is importing the new configuration into production.

## Before you begin

Importing a configuration involves a large number of changes to a Jira instance, so it is a good practice to adopt some safety measures, just in case something goes wrong or not according to expectations.

The most recommended protection is to have a valid backup of the production database from just before the import. This usually requires that the production instance is closed to users. The reason for that is the fact that user operations performed after the backup would be lost if the backup had to be restored.

Close the production instance to users and back it up, either with the Jira XML backup tool, or by using native database tools.

## Import the project configuration

1. In the production instance, open a Jira session as a user with the system administrator permission.
2. In the top-right corner, open the Administration menu, and click **Add-ons**.
3. In the menu on the left, in the PROJECT CONFIGURATOR section, click **Import project configuration**. A page will open, where you can specify which configuration file you want to load, and choose some import options.

The screenshot shows the 'Import project configuration' page in Jira. The navigation menu on the left includes 'ATLASSIAN MARKETPLACE', 'JIRA SUITE UTILITIES', and 'PROJECT CONFIGURATOR'. Under 'PROJECT CONFIGURATOR', 'Import project configuration' is selected. The main content area has a 'Project Configuration File' field with a 'Choose File' button and a file name 'config-dump-...eme.xml.txt'. Below this are several checkboxes for import options: 'Apply changes?' (When ticked, configuration changes will be applied to JIRA), 'Create other projects?' (When ticked, load will create other projects needed by custom field configuration contexts), 'Smart custom field contexts' (When ticked, load will change only those custom field configuration contexts related to projects being imported), 'Try to publish drafts' (Try to publish automatically workflow drafts and workflow scheme drafts created during the import), and 'Continue on errors found in dashboards and filters' (When ticked, load will not stop when an error is found importing a dashboard or filter). A 'Do not load:' dropdown menu is open, showing options like 'Projects (changes)', 'Project specific', 'Versions', 'Components', 'Role members', 'Global', 'Users', and 'Groups'. At the bottom is an 'Import project configuration' button.

This page shows a form with several elements:

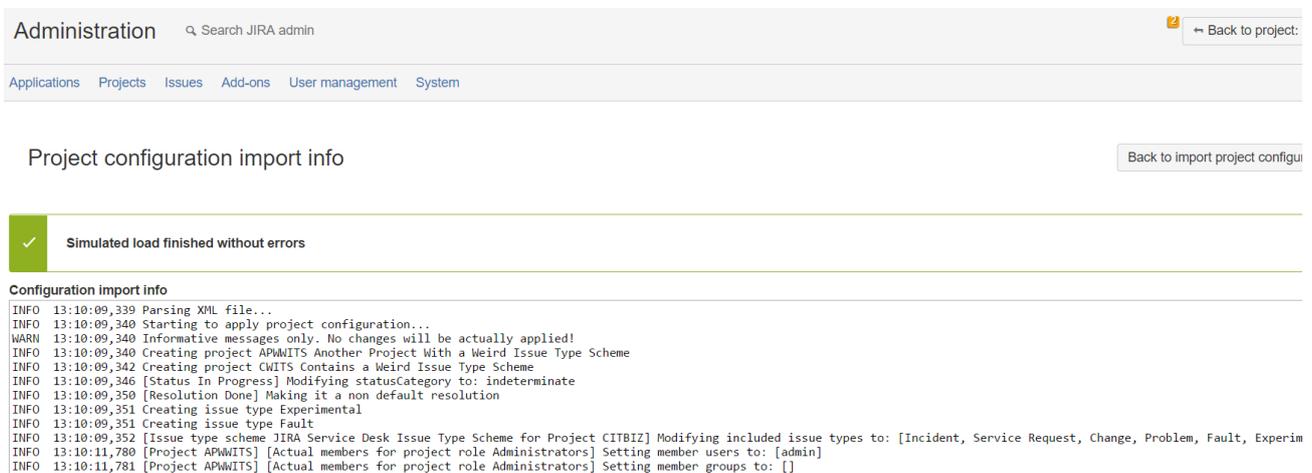
- A notice about warnings before importing configurations (only if you have never launched an import before)
- A button that lets you select an XML file to be imported
- Several other **import options**, which you can select to enable
- A list of object types, which you can select to be ignored during the import

4. Click **Choose File**, and select the XML file you exported from staging. Make sure that the **Apply changes** radio button is cleared. This will allow you to run a simulation without loading the changes.

## 5. Select some import options.

 You can read about each import option [here](#), but let's first see how they are used in most cases:

- **Create other projects** Usually, this option is enabled if you choose to disable the **Smart custom field contexts** option.
- **Smart custom field contexts** Enable this option if the production instance has a large number of custom fields and projects that use them, and you don't want the new configuration to impact other projects. If you'd rather have the custom fields configured exactly the same as in staging, then disable this option.
- **Try to publish drafts** Most users enable this option so that the add-on automatically publishes new versions of workflows and workflows schemes.
- **Continue on errors found in dashboards and filters** Enable this option if you're importing a large number of filters and dashboards that are not controlled by the admin /development teams (they were authored by the users.) Otherwise, you can disable it.

6. Click **Import project configuration** at the bottom of the page. This will launch a simulated configuration load. You will then see a page with results.


Administration  [Back to project:](#)

Applications Projects Issues Add-ons User management System

Project configuration import info [Back to import project configu](#)

 Simulated load finished without errors

**Configuration import info**

```
INFO 13:10:09,339 Parsing XML file...
INFO 13:10:09,340 Starting to apply project configuration...
WARN 13:10:09,340 Informative messages only. No changes will be actually applied!
INFO 13:10:09,340 Creating project APWILTS Another Project With a Weird Issue Type Scheme
INFO 13:10:09,342 Creating project CWILTS Contains a Weird Issue Type Scheme
INFO 13:10:09,346 [Status In Progress] Modifying statusCategory to: indeterminate
INFO 13:10:09,350 [Resolution Done] Making it a non default resolution
INFO 13:10:09,351 Creating issue type Experimental
INFO 13:10:09,351 Creating issue type Fault
INFO 13:10:09,352 [Issue type scheme JIRA Service Desk Issue Type Scheme for Project CITBIZ] Modifying included issue types to: [Incident, Service Request, Change, Problem, Fault, Experim
INFO 13:10:11,780 [Project APWILTS] [Actual members for project role Administrators] Setting member users to: [admin]
INFO 13:10:11,781 [Project APWILTS] [Actual members for project role Administrators] Setting member groups to: []
```

This page shows a simulation of all operations that would be performed on the production instance after importing the configuration file. Since it's only a simulation, none of these changes have actually been applied. Review this information, and check if the changes are aligned with your expectations.

If everything is fine, you can launch the actual import. Go back to the **Import project configuration** page, again select the XML file, and enable the **Apply changes** option. Finally, click **Import project configuration** at the bottom of the page. You will obtain a similar results page with the trace of all changes that have been applied to the production instance.

Let us know what you think

[Feedback](#)

# Verifying results of promotion

## Check results of promoting changes into production

After completing the import, you can perform a quick review of the results. First of all, you can examine the import trace that Project Configurator created during the real import. Check if there are any errors that might have interrupted the import, or errors regarding the import of filters or dashboards that wouldn't stop the import, but might have an impact on a specific filter or dashboard.

You can also run some quick tests. All the advice about tests in staging given on [this page](#) applies also to the production instance. In fact, you could use a subset or all of these tests now.

## Open the production instance to users

Finally, if these tests show that the new configuration is working as expected, the only thing left to do is opening production instance, so that every user can log in and start the work.

Let us know what you think

[Feedback](#)

# Troubleshooting

If you encounter any problems while exporting or importing your configuration changes, check the following resources:

- [Export contents of an error page explained](#)
- [Export most frequent errors](#)
- [Import contents of an error page explained](#)
- [Import most frequent errors](#)
- [Issue tracker](#) issues with their analysis and solutions
- [Support channels](#)

Let us know what you think

[Feedback](#)

# Migrating projects to another Jira instance

**i** These steps are for migrating projects between different instances of Jira Server, and require a 3rd party app [Project Configurator for Jira](#). If you're migrating from Server to Cloud, see [Jira Server to Cloud migration resources](#).

Let's assume you have two instances of Jira a source, and a target. In the source instance, you have several projects and users working on these projects. You'd like to transfer the projects to the target instance along with the configuration, issues, and attachments. Ideally, after the migration, your users wouldn't even notice the change. Here are a couple of possible scenarios:

- **Expanding a small project** the source instance is operated by a group or department within a bigger organization. A project is started there as an experiment. After some time, however, this project has expanded to other groups, and you want to move it to the target, corporate instance of Jira.
- **Merging Jira instances** a company acquires another company. Both have their own Jira instances and decide to consolidate them into a single instance. Merging one Jira with another requires migrating all projects.
- **Balancing the use of licenses** a company runs two instances of Jira, one with 500 user licenses, and another with 10,000 user licenses. If, in the bigger instance, only 8,000 users are actively working, you can move some projects from the smaller instance to improve the balance.

## Project Configurator for Jira

This guide describes how to migrate your projects with [Project Configurator for Jira](#). It offers an export function that packs your configuration, issue data, and attachments into a single ZIP archive. You can then transfer it to the target server, and import your projects from the ZIP archive.

# Preparing for migrating projects

**i** These steps are for migrating projects between different instances of Jira Server, and require a 3rd party app [Project Configurator for Jira](#). If you're migrating from Server to Cloud, see [Jira Server to Cloud migration resources](#).

## Versions

We recommend that Jira versions on the source and target instances are the same. If that's not the case, you should upgrade one of the instances.

However, in some cases, you can work around this limitation.

### Migrating between different Jira versions

- If the Jira version on the target instance is earlier than on the source instance, you'll need to upgrade the target instance.
- If the Jira version on the source instance is earlier than on the target instance, you have two options:
  - Upgrade the source instance.
  - Migrate by using a staging server, following these steps:
    1. Clone the target instance to a staging server.
    2. Take an XML backup of the source instance.
    3. Restore the backup on the staging server. This will erase all content on the staging server, and upgrade the migrated data to a later version.
    4. Use Project Configurator for Jira to migrate your projects from the staging server to the target instance using the procedure described in this guide.

## Disk space requirements

When migrating, you'll export your projects into a ZIP archive that will contain the following information:

- Configuration of the exported projects
- Issues in those projects (including attachments, comments, worklogs, history, and so on)

Before the ZIP archive is created, its components are assembled in a temporary directory, which requires even more space than the final ZIP archive. We recommend that the size of the temporary directory is around 4 times the size of the final ZIP archive.

You can use formulas to calculate the approximate size of the ZIP archive. To do this, you'll need the following information:

- The size of an XML backup of your database.
- The total number of projects in your instance, and the number of projects you want to migrate.
- The size of attachments for the migrated projects. Attachments for each project are stored in separate directories in `<Jira-home-directory>/data/attachments`, each directory is named after the project's first key. This path is the default directory, you can check it in Jira by going to  **> System > Attachments**.

1. Use the following formula:  
 $size\ of\ the\ database * migrated\ projects / all\ projects * 1.2$

For example, if the size of your DB is 500 MB, and you want to migrate 5 projects out of 20:  
 $500\ MB * 5 / 20 * 1.20 = 150\ MB$

2. Next, check the size of attachments for the migrated projects, and use another formula.  
 $size\ of\ the\ attachments / 4$

For example, if the size of your attachments is 300 MB:  
 $300\ MB / 4 = 75\ MB$

The estimated size for the final ZIP archive, using the above examples, would be 225 MB. The

disk space for the temporary directory must be around 1 GB, because it must be 4 times the size of the ZIP archive.

Provide the same disk space both on the source and the target instance, because the ZIP archive will be uncompressed on the target instance before being imported to Jira.

## Licenses

Make sure that you have enough licenses for your users in the target instance. If you don't, you'll need to provide more, or deactivate some users after the migration. For more information, see [Create, edit, or remove a user](#).

## User management

User management during the migration depends on your specific environment, however, you should consider the following rules:

- Users are mapped from the source to the target instance by usernames (field `Username` in Jira). If a single person has two different usernames in both instances, you should make them consistent:
  - Rename one of the users, or
  - If the user is managed in an external directory with LDAP, which has some attribute that is the same for both usernames, you can [configure it to act as 'username'](#).
- Users managed in external directories can be migrated with the following methods:
  - Tools specific to external directories (e.g. replicating users across LDAP instances). In this case, you should migrate your users before migrating your projects. They might be referenced in some parts of the projects' configuration (e.g. permissions).
  - Project Configurator for Jira. For more information and restrictions, see [Specific information for some object types](#).
  - A combination of both.

These considerations also apply to group management.

## Matching properties of both instances

Make sure the following properties are the same for both instances:

- timezone
- locale (especially the rules that define how numbers and dates are formatted into strings)
- the maximum size for text fields (the size in the target instance can be larger than the size in the source instance)

For more information about setting the size limit for text fields, [click here](#).

## Names of configuration objects

The migration process will treat configuration objects with the same names in both instances as representing the same thing. It's mostly the case, but sometimes two objects with the same name might be completely different. The configuration coming from the source will overwrite the configuration in the target because it's assumed that the source represents the newer configuration you want to implement. To avoid losing some items:

- Review the names of [globally available configuration objects](#) in both instances
- If there are coincidences, check if those name actually represent the same object.
- If needed, rename an object in one of the instances to avoid overwriting it with wrong data.

Project Configurator offers some features that might help you with this issue:

- [Import conflict detection](#) it produces a summary report of all configuration objects in the target instance that will be mapped to objects coming from the source instance. The report also shows where the duplicated objects are used in the target instance.
- ["Used by" report](#) it shows where the configuration objects are used. This is useful if you need to rename, change, or delete any of these objects.

### Plugins defining custom field types

Any plugin that defines custom field types in the source instance must also be installed in the target instance with the same version.

### Plugins defining workflow extensions (conditions, validators, post-functions)

Any plugin that defines workflow extensions in the source instance must also be installed in the target instance, either in the same or a newer version.

- Workflows from the source instance will be migrated to the target instance. If the required plugins are not there, the workflows might not work.
- If you don't install the plugins, the import might fail because the "create transition" action of the corresponding workflow is run for every issue imported into the target instance.

### Migrating Agile boards

Migration of Agile boards is supported, but if you're still on Jira 6, make sure you have Jira Agile 6.7.7, or later. For earlier versions, you won't be able to migrate the boards.

### Migrating sprint and ranking data

Migrating sprint and ranking data is supported only for Jira Software. The data from Jira Agile will be ignored.

### Language and locale

The source and the target instance must be installed with the same language and locale. Jira Software creates its fields with different names depending on the language settings. This happens even if one of the instances uses English US, while the other English UK (e.g. Epic Color and Epic Colour). For more information, see [this issue](#).

### Duplicate and obsolete rank fields

In some cases, an upgrade of Jira Software creates fields of type Rank, which are called Rank (Obsolete). If the projects you want to migrate use these fields, you'll probably encounter errors during the migration. These custom fields are usually locked, which means they must be created or configured by Jira Software (Project Configurator will not create them in your target instance). As a result, your projects in the source and target instance will be using different sets of fields, which breaks the migration.

To avoid these issues, complete the following steps:

1. Verify that Jira Software is installed on both the source and the target.
2. If there are duplicate or obsolete Rank fields in the source instance, remove these fields and the related rank values. For more information, see [Jira KB article](#) and [JSW-13098](#).

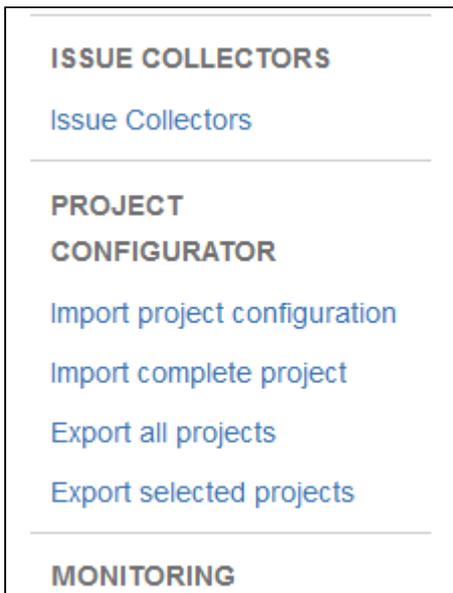
# Exporting projects from the source instance

**i** These steps are for migrating projects between different instances of Jira Server, and require a 3rd party app [Project Configurator for Jira](#). If you're migrating from Server to Cloud, see [Jira Server to Cloud migration resources](#).

Export your projects from the source instance.

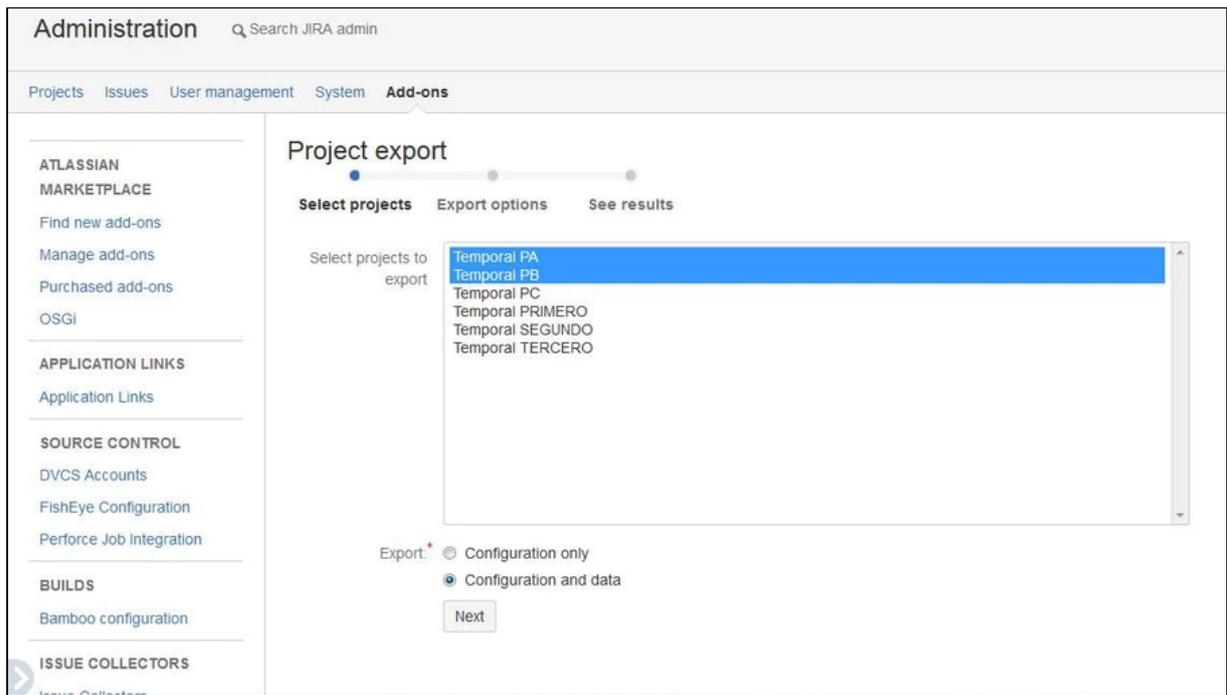
## Starting the export

1. Log in to the source Jira as an administrator.
2. Go to the add-on administration page.
3. In the menu on the left, in the PROJECT CONFIGURATOR section, click **Export selected projects**.



4. Select the projects that you want to export (to select more than one, use the **CTRL** key in Windows, or **COMMAND** key in macOS.)

5. Select **Configuration and data** below the list of projects, and click **Next**.



## Export options

Next, you will choose export options.

When migrating projects, the most important options are related to exporting custom fields and users.

- **Filtering custom fields** Select this option to migrate only the custom fields that are used by your exported projects. It'll save you time.
- **User/group export options** Choose a way to export your user data:
  - **Full export:** This is the default option. If a username is anywhere in the exported configuration, the add-on will try to find and add it to the exported items. If the user is not valid, the export will halt with an error.
  - **Ignore invalid users:** If the username is not valid, the user will not be added to the exported items. The username will still appear as component or project lead, and will be a part of different schemes in the exported file. In these cases, however, the export will not halt. Currently, the add-on can detect an invalid username in two cases:
    - When it can't find a user with that name
    - When the user has an invalid email address (it does not conform to the pattern "X@Y" where X and Y are non-empty strings)
  - **Do not export:** No users will be exported.

For more information about the export options, see [Selecting export options](#).

## Exporting filters, dashboards and Agile boards

As for filters and dashboards, it's enough if you select one of the following options:

- **Shared with exported projects** export filters/dashboards that are used by your exported projects.
- **With all users or with exported projects** export filters/dashboards that are used by your exported projects, or all users.

For Scrum and Kanban boards, you'll probably export only those boards that are associated with the exported projects (Associated to exported projects).

## Export summary

When the export is complete, you will be redirected to a summary of the export that shows the location of the exported ZIP archive. You will need it in the next step.

### Export summary

```
Launching export...OK
Exporting configuration...OK
Exporting project data...OK
Exporting attachments...OK
Exported attachments:
  - Project PA: 0 attachments
  - Project PB: 0 attachments
Compressing files...OK

Export file available at D:\nuevo\jira-project-config-plugin\target\jira\home
\export\projectconfigurator\project-dump_BP8Q-WXN6-SKX3-NB5M_2_PROJECTS.zip
```

# Running a test migration

**i** These steps are for migrating projects between different instances of Jira Server, and require a 3rd party app [Project Configurator for Jira](#). If you're migrating from Server to Cloud, see [Jira Server to Cloud migration resources](#).

Before you migrate your projects to the original target instance, you can establish a staging environment, which will be a copy of your Jira, and test the migration.

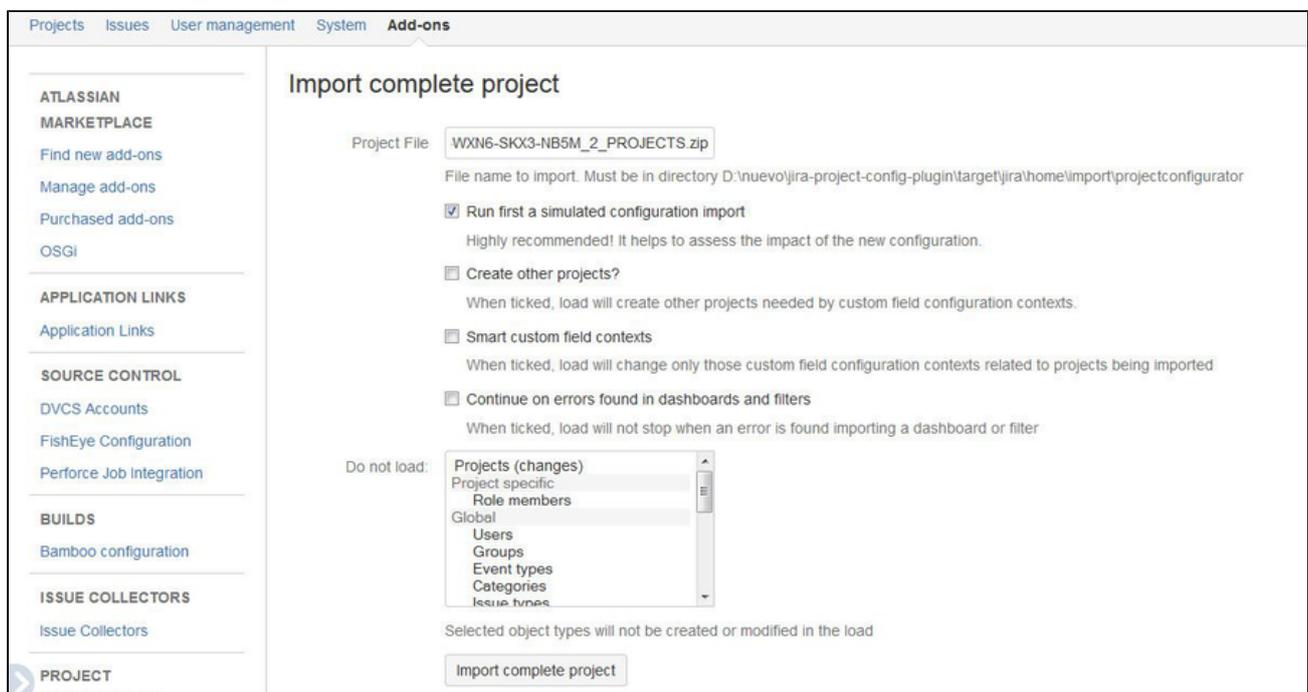
## Before you begin

[Create a staging environment](#) for testing the migration.

## Importing

1. In your test instance, go to `<Jira-home-directory>/import`, and create a directory called `projectconfigurator`.
2. Copy the exported ZIP archive to the `projectconfigurator` directory.
3. Log in to Jira as an administrator, go to the add-on page, and click **Import complete project**.
4. In the Project File field, enter the name of the exported ZIP archive. Make sure to include the `.zip` extension.
5. Select **Run first a simulated configuration import**.
6. Click **Import complete project**.

The following image shows the relevant options for importing a complete project.



## Import options

The default values are sufficient for most environments. However, if you'd like to check other import options, take a look at the following definitions:

- **Run first a simulated configuration import** this option is recommended, and gives you a chance to review changes to the configuration before applying them.

- **Smart custom field contexts** enable this option if some custom fields from the source instance have the same type and name as different custom fields already existing in the target. For more information, see [Smart custom field contexts](#).
- **Continue on errors found in dashboards and filters** often, users create and maintain filters and dashboards without any supervision or help from the administrators. If that's the case for your team, enable this option, so that the migration process skips any inconsistencies or errors in the configuration of filters and dashboards

## Summary

After your projects are imported to the test instance, you'll see a summary of the import.

Next, a page will be displayed showing the configuration changes that will be applied to the test instance. The content of this page is similar to the simulated import. The difference is that here you must verify and accept the proposed changes.

✓
**Simulated load finished without errors**

⚠
**Before continuing...**

Please review changes to be made to configuration before clicking "Next" to continue

Next

### Configuration import info

```
INFO 14:35:04,105 Parsing XML file...
INFO 14:35:04,109 Starting to apply project configuration...
WARN 14:35:04,109 Informative messages only. No changes will be actually applied!
INFO 14:35:04,110 Creating project PA Temporal PA
INFO 14:35:04,110 Creating project PB Temporal PB
```

## Drafts for workflows and workflow schemes

In most cases, Project Configurator automatically creates and publishes the required drafts. This happens in the background, and you won't notice it unless you examine the import trace.

In some cases, however, the add-on will not be able to publish the drafts, either because your input is required to map issues to the new status, or the structure of a new workflow is not compatible with the old one. The add-on will stop, and you'll need to complete extra steps to continue. For more information, see [Optional stop for publishing drafts](#).

# Validating the test migration

**i** These steps are for migrating projects between different instances of Jira Server, and require a 3rd party app [Project Configurator for Jira](#). If you're migrating from Server to Cloud, see [Jira Server to Cloud migration resources](#).

When the test migration is complete, you'll need to review the results and traces before you proceed to migrate your projects to the target instance. You should also check what changes, if any, are needed, and apply them to your source instance.

## Import results and import trace

When reviewing the import trace, keep in mind that:

- **Data import** is carried out in a sequence for each of the migrated projects.
- **Warnings** they do not stop the import but might cause some pieces of information to be skipped (e.g. values from some custom fields).
- **Errors** these will stop the data import for a project. If you're migrating several projects and some of them fail, retry the import. Project Configurator for Jira will detect the already migrated projects and omit them. If you want to retry the import for a project that already has some data (any issue, version, or component), delete this project at the target instance, and only then retry the import. Otherwise, this project might be detected as the already migrated one.

### Project data import results

[Back to import complete project page](#)

 **Import of data and attachments finished**

#### Import results

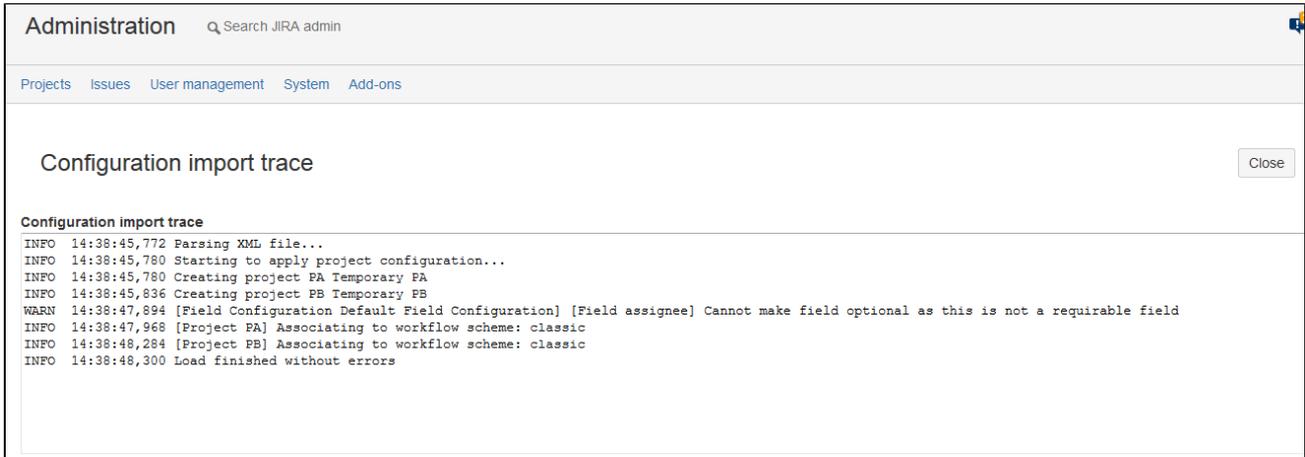
```
PROJECT: Project DL Frontend
Time: 318
Number of created users: 0
Number of created issues: 3
Number of created attachments: 0
PROJECT: Security
Time: 29
Number of created users: 0
Number of created issues: 0
Number of created attachments: 0
```

#### Import trace

```
Starting data import for project with key: PDLF
INFO 12:27:02,053 Project Import: Parsing the backup file 'D:\other\support-and-debug\amps-standalone\target\container\tomcat7x\cargo-jira-home\temp\com.awnaba.projectconfigurator1475144786701\data\data.zip' to obtain a Backup Overview.
INFO 12:27:02,504 Project Import: Backup Overview was successfully extracted from 'D:\other\support-and-debug\amps-standalone\target\container\tomcat7x\cargo-jira-home\temp\com.awnaba.projectconfigurator1475144786701\data\data.zip'.
INFO 12:27:03,146 Project Import: Mapping the backed up data to data in the current system, and validating the mappings...
INFO 12:27:03,190 Project Import: No validation errors were found and the import can continue.
INFO 12:27:03,195 Starting project import for project 'PDLF'.
INFO 12:27:03,195 Creating missing users. Attempting to create 0 users.
INFO 12:27:03,198 Finished creating missing users. 0 users created.
INFO 12:27:03,205 Creating the issues.
INFO 12:27:03,237 Finished creating the issues.
INFO 12:27:03,315 Creating the issue-related data.
INFO 12:27:03,367 Finished creating the issue-related data.
INFO 12:27:03,371 Creating the change item data.
INFO 12:27:03,378 Finished creating the change item data.
INFO 12:27:03,380 Creating the attachments.
INFO 12:27:03,386 Finished creating the attachments.
```

## Configuration changes

To review configuration changes, select **Click here to see configuration import trace**. It will show configuration changes that were applied to your test instance before the import.



## Validate results of test import

After finishing the test import, examine the test instance and validate the migration results. Some suggested checks:

After completing the test import, examine the test instance, and validate the migration results. We recommend to check the following items:

- The projects were migrated, and can be viewed in Jira.
- The projects have the same number of versions and components as in the source instance.
- Role members are the same. If the number of projects is too big to check that, verify a sample of roles and projects.
- The number of issues and attachments is the same.
- Review a sample of migrated issues, focusing on the following items:
  - General issue data (including custom field values)
  - Sequence of comments
  - Change history
  - Worklogs
  - Attachments (number, name, and content)

## User acceptance tests

Finally, select a group of users, so they can work on and test the migrated projects. They should be using the same account they had in the source instance. Check the following items:

- Can you view, and edit issues?
- Can you launch transitions?
- Do you have access to reports, filters, dashboards, or Agile boards?
- Can you access the same information you were using in the source instance?

## Record any changes required for a successful migration

During the test migration, you might need to make some changes to the source and target instances, or even files created during the export process, so that the final migration completes successfully. It's also possible that the reviews and tests described on this page detect some other issues, which require changes in both instances. It's important that you record all these changes and fixes so that you can repeat them before the final migration.

# Importing projects into the target instance

**i** These steps are for migrating projects between different instances of Jira Server, and require a 3rd party app [Project Configurator for Jira](#). If you're migrating from Server to Cloud, see [Jira Server to Cloud migration resources](#).

After you've successfully tested and validated the migration, you can migrate your projects to the target instance. The whole process will be just the same as the test migration.

## Before you begin

Back up the target instance, so you can restore it in the case of any problems.

## Migrating

1. Lock the source instance, so that your users can't make any changes.
2. [Export your projects](#) from the source instance.
3. [Import your projects](#) to the target instance.
4. [Validate the migration results](#).
  - If you're happy with the outcome, open the target instance to your users.
  - If something is not right, remove the migrated projects from the target instance, and retry the migration.

# Troubleshooting the migration

**i** These steps are for migrating projects between different instances of Jira Server, and require a 3rd party app [Project Configurator for Jira](#). If you're migrating from Server to Cloud, see [Jira Server to Cloud migration resources](#).

If you encounter any problems with the migration, check the available resources and the list of known issues.

## Resources

If you encounter any problems while exporting or importing your configuration changes, check the following resources:

- [Export contents of an error page explained](#)
- [Export most frequent errors](#)
- [Import contents of an error page explained](#)
- [Import most frequent errors](#)
- [Issue tracker issues with their analysis and solutions](#)
- [Support channels](#)

## Known issues

### Custom field context applies only to some issue types

An import fails with the following message:

```
The custom field 'XXXX' in the backup project is used by issue types 'AAAA, BBBB' but the field with the same name in the current Jira instance is not available to those issue types in this project.
```

This is a [known issue](#). To work around it, make the custom fields available to all issue types:

1. In the target instance, [configure the custom fields](#) to be available to all issue types.
2. Rerun the import, but choose to ignore custom fields in the configuration import. Otherwise, the custom fields will be imported from the source, overwriting your changes in the target instance.
3. Optional: After a successful import, you can restore these custom fields to their original configuration (restricted to some issue types).

### Validation errors with no extra information

The following warnings appear when a value of a text field exceeds the maximum text field size in the target instance. For more information about changing this limit, see [Preparing for migrating projects](#).

```
WARN 11:00:56,968 The attachment 'MMMMMMMM.zip' does not exist at '/tmp/com.awnaba.projectconfigurator1493567880435/data/attachments/XXXXXXXXXXXX'. It will not be imported.

WARN 11:00:56,968 The attachment 'screenshot-1.png' does not exist at '/tmp/com.awnaba.projectconfigurator1493567880435/data/attachments/YYYYYYYYYYYY'. It will not be imported.

WARN 11:00:56,968 The attachment 'screenshot-1.png' does not exist at '/tmp/com.awnaba.projectconfigurator1493567880435/data/attachments/ZZZZZZZZZZZZZZ'. It will not be imported.

INFO 11:00:56,969 Project Import: Validation errors were found. The import cannot continue.
```

### Problem with migrating values for the Time in Status custom field

The import trace shows the following warning:

```
WARNINGS: Unable to import custom field 'Time in Status'. The custom field type does not support project imports.
```

Since it's only a warning, it will not stop the import, but the values for this custom field will not be imported. There's no need to worry about that because these values will be automatically recalculated.

# Merging Jira instances

**i** Tasks described on this page are not native to Jira. To complete them, you'll need to install a 3rd party add-on [Project Configurator for Jira](#).

## Overview

Merging Jira instances requires migrating all projects, issues, dashboards, etc. from one instance to another without deleting or changing the existing content in the target instance. After the merge, users are expected to resume their work in the target instance, working with the same set of data.

Here are some reasons why you might want to merge your Jira instances:

- A company purchases another company, both of them are using Jira and want to merge their two instances into a single one.
- A company has several instances of Jira in different departments and wants to consolidate all of them into a single, corporate Jira instance.

## Steps

With some differences explained on this page, **merging Jira instances** is the same as moving all projects from the source instance to the target instance. You should use the steps described in [Migrating projects to another Jira instance](#), and then proceed with the details described here.

### Are all configuration objects migrated?

With some exceptions, where [export options](#) allow the user to control what objects will be included in the export, all configuration objects that are *directly or indirectly required by a project* are exported. Objects that are not used by any project, like inactive workflows, screens, or schemes are not transferred.

### Exporting all projects

To export all projects from a Jira instance:

1. Log in to the source Jira as an administrator.
2. Go to the add-on administration page.
3. In the menu on the left, in the PROJECT CONFIGURATOR section, click **Export all projects**.

---

**PROJECT  
CONFIGURATOR**

- [Import project configuration](#)
- [Import complete project](#)
- [Export all projects](#)
- [Export selected projects](#)

### Best export options

These options will be most useful when exporting your projects:

**Filtering custom fields** You can either export all custom fields or only those that are used by your projects. In the latter case, custom fields that are not used by any project will be omitted.

**User export options** and **Group export options** see the discussion on user management [here](#).

## Exporting filters, dashboards and Agile boards

When merging instances, you'll need to transfer all filters, dashboards, and boards.

- For filters, select **All filters (shared or private)**.
- For dashboards, select **All dashboards (shared or private)**.
- For Scrum and Kanban boards, select **All Scrum and Kanban board**.

## What to do if the ZIP file is too big

If exporting or importing a ZIP file with all projects is taking too long, or if the ZIP file is too big, try splitting the process into smaller chunks. For example, if the source instance has 50 projects, migrate them in 5 batches, each containing 10 projects. You can expect the import of the first batch to be more complex than the rest because it also needs to apply all the configuration. You might also want to postpone the import of filters, dashboards, and Agile boards until the last batch when all projects are already available in the target instance.

## Shut down the source Jira instance

To prevent users from logging in to the source instance and creating new data there after the migration, you should shut down this instance, or at least leave it in the read-only mode. Keeping it in the read-only mode might be useful for Jira admins who could use it in case any data inconsistency for the merged projects is detected after the merge.

# Migrating data with Botron

An add-on for Jira prepared by Botron Software helps you transfer your configuration, projects, and all associated data from one Jira to another. Read about possible scenarios and choose the one you need.

## Promoting Jira configuration

[Learn more](#) about moving your changes from development through staging to production to make sure they're all checked up. By doing so, you protect your production environment from any mistakes or unevaluated changes.

## Consolidating Jira instances

[Learn more](#) about consolidating multiple Jira instances into a single one.

## Migrating Jira projects

[Learn more](#) about migrating your projects, along with all relevant data, from one Jira instance to another.

# Promoting Jira configuration from development to production

 Tasks described on this page are not native to Jira. To complete them, you'll need to install a 3rd party add-on [Configuration Manager for Jira](#).

## Overview

This document describes best practices for promoting Jira configuration from the development to production environment. You can use Botron's Configuration Manager add-on to effectively test your desired changes prior to rolling them out in the production environment.

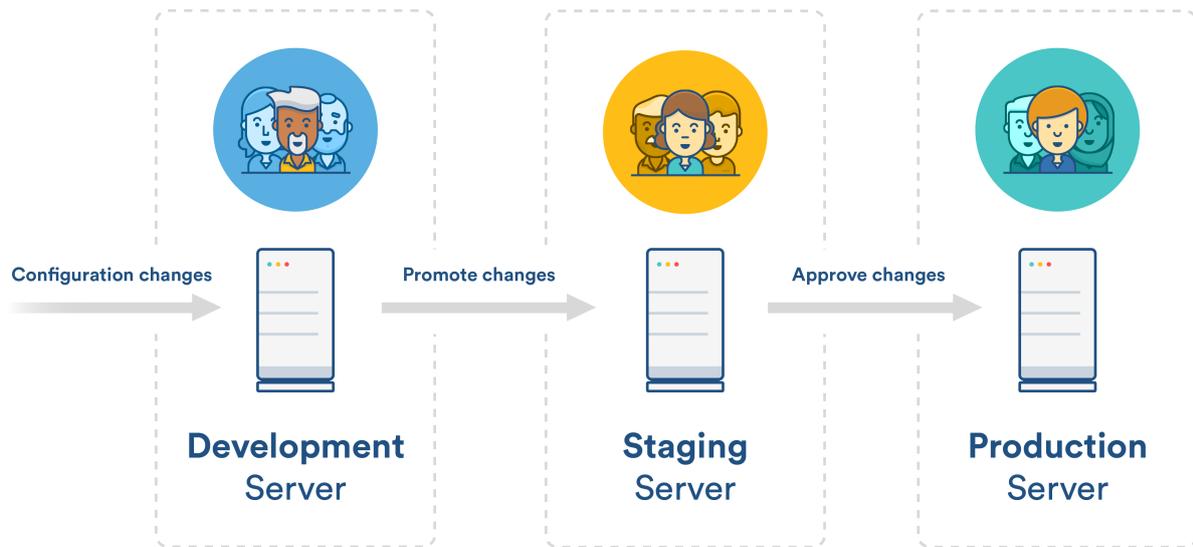
- [Architecture Strategy Recommendations](#)
- [Planning](#)
- [Stages of three-tier architecture strategy](#)
- [Moving add-on data](#)
- [Limitations & workarounds](#)
- [Common issues](#)
- [Frequently Asked Questions](#)
- [Need help?](#)

## Definitions

For this document, we'll assume the following definitions:

- **Development** A free-for-all one or many environments where users can play with cutting-edge or risky changes.
- **Staging** A pre-production environment, where the systems administration team can establish exact procedures prior to rollout. The staging should be a clone or close replica of the production environment.
- **Production** Your live instance, expecting minimal downtime and well-tested changes.
- **Configuration snapshots** are created using the [Configuration Manager for Jira](#) add-on and represent the state of your Jira [configuration objects](#) and their relations to each other at a given point in time. There are two types of configuration snapshots:
  - **Project snapshot** Contains the configuration of a number of selected projects (with their schemes, workflows, fields, etc.)
  - **System snapshot** Contains the entire configuration of a single Jira instance (projects, workflows, schemes, screens, etc.)

## Architecture Strategy Recommendations



If Jira is a critical system, we recommend a 3-tier architecture strategy consisting of **development**, **staging**, and **production** environments.

If Jira is not a critical system, you can use a 2-tier strategy with **development** and **production**.

## Planning

### Prepare environments (hardware)

Staging should be as close as possible to the hardware used for the production server. Although the development can be any type of hardware, the general guideline is that it is as close as possible to production too.

### Keep the staging in sync

The staging environment is used as final rehearsal prior to introducing changes in production. To ensure that the rehearsal is accurate and to eliminate potential surprises during the actual deployment, the staging should be an identical replica of the production environment.

Depending on the type of hardware used for production, there are several options for keeping the staging in sync:

- **Physical hardware** Keeping staging in sync can be accomplished by either cloning the storage used for the database and filesystem, or simply following the Jira [Backup/Restore](#) procedure.
- **Virtual machine** Creating a snapshot of the production, and creating a new virtual machine from this snapshot.
- **Configuration Manager** A unique feature allows the staging server configuration to be [restored](#) with the production.

### Configuration Manager for Jira

Botron's [Configuration Manager for Jira](#) add-on needs to be [installed](#) on each Jira server.

### Tracking

It is recommended that every change to Jira setup and configuration is tracked via **change request ticket**. The change request ticket should follow the chosen promotional path and include the initial user request, as well as additional information as it progresses through the lifecycle configuration snapshot, change and impact analysis, duration, etc.

### Disruptive changes

A recommended IT practice is that all non-emergent fixes should be introduced in a defined cadence, e.g. each Friday afternoon from 1 to 3 PM. The changes introduced can be segmented into two groups **disruptive** and **non-disruptive**. The former can be introduced only in defined maintenance windows when the user access is limited.

### Communication strategy

A communication strategy is designed to inform Jira users about the changes that will be introduced this information includes the exact date of the changes roll-out, as well as a comprehensive summary. A link to the change request ticket might also be included.

The communication strategy can be conducted via email, notification banners, or other standard means of internal communications. A recommended practice is to inform the users at least 5 times:

- 1 week prior to the changes being introduced
- At the beginning of the business day
- 1 hour before
- Notification right before
- After the changes are introduced, a message indicating success/failure.

### System health

The system health of all 3 servers should be assessed, and any [Integrity Check](#) errors should be resolved.

## Stages of three-tier architecture strategy

In this section, we'll walk you through the process of rolling out changes for Jira production environment with a 3-tier architecture strategy. The illustrated processes require the installation of the [Configuration Manager for Jira](#) add-on, which enables you to [create and deploy configuration snapshots](#) between the different environments. The add-on should be installed on each environment. For more information regarding licensing visit the FAQ section.

### First stage (development)

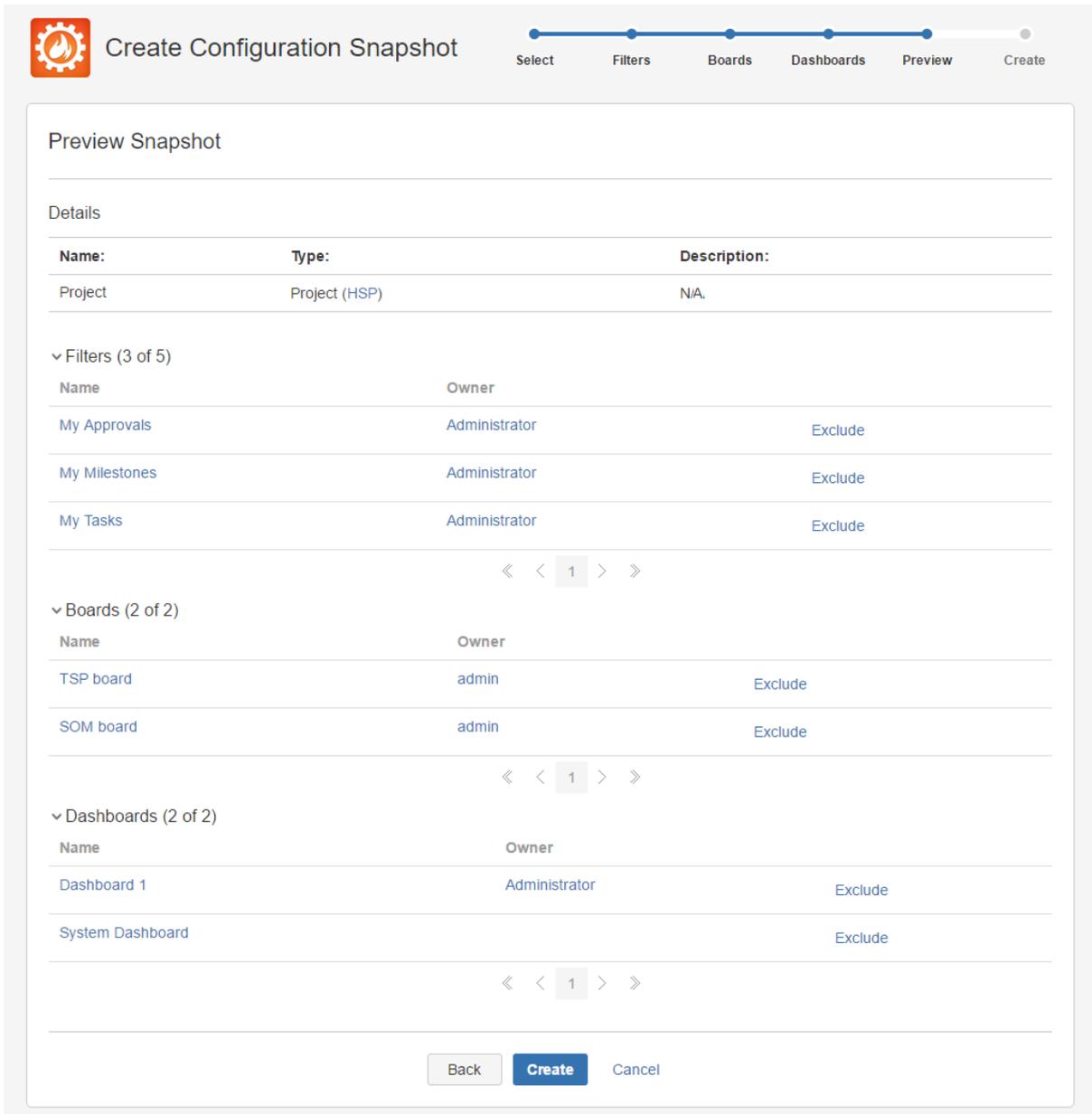
Overview of the first stage:

- **Environment:** Development environment
- **Goals:** Develop new project configuration; allow user acceptance testing; create **configuration snapshot** that can be **promoted** to the **staging** environment.
- **Users:** Business users or administrators. Open to anyone with expertise in Jira configuration.
- **Tools required:** [Configuration Manager for Jira](#)

The first stage of this process is conducted in the **development environment**. It typically involves the following **users**: business users or administrators, anyone with an expertise in Jira configuration.

The **steps** of this stage are as follows:

1. **Develop new project configuration** a project configuration can contain several configuration elements, e.g. workflows, custom fields, screens, etc.
2. Once the new configuration is in place, **user acceptance testing** is performed to ensure proper configuration.
3. [Create configuration snapshot](#) that can be **promoted** to the **staging** environment.



**Create Configuration Snapshot**

Preview Snapshot

Details

Name:	Type:	Description:
Project	Project (HSP)	N/A.

Filters (3 of 5)

Name	Owner	
My Approvals	Administrator	Exclude
My Milestones	Administrator	Exclude
My Tasks	Administrator	Exclude

Boards (2 of 2)

Name	Owner	
TSP board	admin	Exclude
SOM board	admin	Exclude

Dashboards (2 of 2)

Name	Owner	
Dashboard 1	Administrator	Exclude
System Dashboard		Exclude

Back Create Cancel

4. **Download the configuration snapshot**(if your Jira configurations are being tracked via tickets, the snapshot should be attached to the appropriate tickets.)

✔ Configuration Manager for Jira will include **only** configuration objects referenced directly or indirectly by the project. If you want to add custom fields to your configuration snapshot, certain conditions must be met. For more information on adding custom fields to your configuration snapshot, [click here](#).

## Second stage (staging)

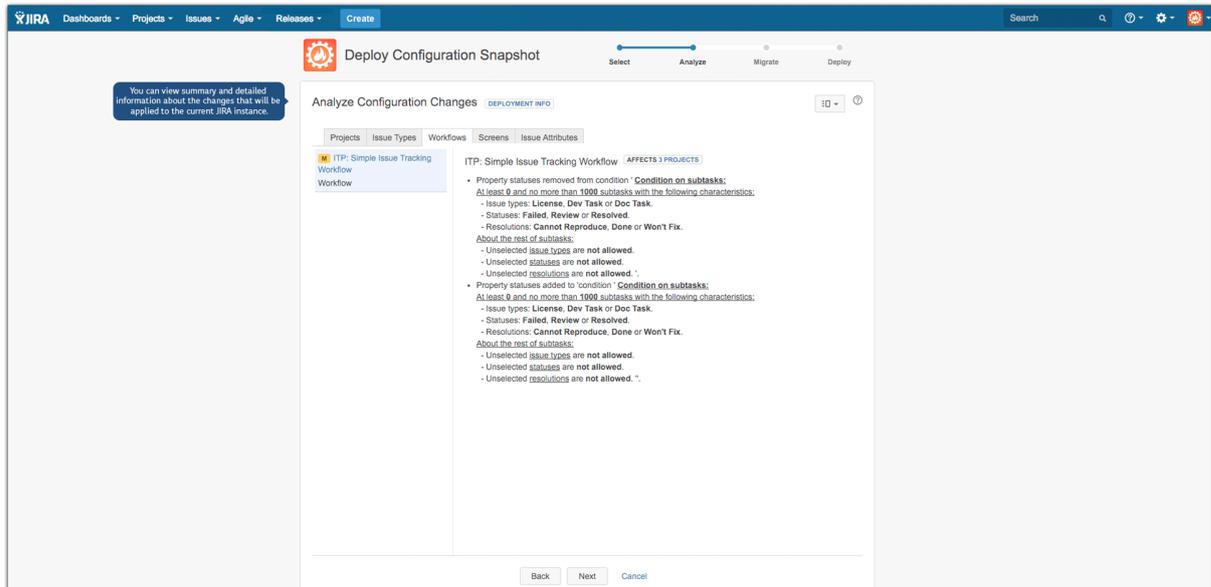
A high-level overview of the second stage:

- **Environment:** Staging environment
- **Goals:** Validate proposed configuration changes; assess changes and impact; prepare communication plan; estimate the required downtime, and clarify production rollout procedure.
- **Users:** Jira System administrators
- **Tools required:** [Configuration Manager for Jira](#)

The second stage of this process is conducted in the **staging environment** and involves Jira system administrators who perform a comprehensive validation to ensure that the introduced changes won't impact other projects. There is a difference between **change** and **impact**. For example, a **change** in the name of a workflow status (from Done to Accepted) might interfere with the company policy or break JQL filters used for reporting. A typical example of an **impact** is when a change to the notification scheme is introduced that change will likely impact other projects in production.

The **steps** of this stage are as follows:

1. **Deploy configuration snapshot** and validate proposed configuration changes.



2. It is crucially important to validate the proposed new configuration for **changes and impact**. If the result is negative go back to Development.
3. Communicate any changes to your users prior to officially introducing them.
4. Estimate the required downtime and clarify production roll-out procedure by measuring the required deployment time and updating your ticket.

### Third stage (production)

A high-level overview of the third stage:

- **Environment:** Production environment
- **Goal:** Roll-out configuration changes to production
- **Users:** Jira System administrators
- **Tools required:** [Configuration Manager for Jira](#)

The third and final stage of this process includes rolling out your configuration changes to production. After ensuring that the results of the change and impact analysis of the previous stage don't show any conflicts or undesired impact, move forward with the deployment to the production environment.

The **steps** of this stage are as follows:

1. Execute the communication plan required.
2. Create a configuration snapshot for backup.
3. Limit user access changes to production servers are done during **maintenance windows** when the user access is limited (this is not a strong requirement but rather a recommended best practice.)
4. **Deploy configuration snapshot**.
5. Review audit logs for any warnings.

Administration
Search JIRA admin

Projects
Add-ons
User Management
Issues
System
Configuration Manager
Audit Log

Snapshots

Deploy

---

**Integrity Check**

Audit

---

Get Started

## Configuration Integrity Check

Run Integrity Check ?

**Failure**  
180 configuration elements were checked and 40 errors were found. See details in the table below.

**Unknown errors were detected**  
Send **support zip** to [support@btronsoft.com](mailto:support@btronsoft.com), so we can assist you in resolving this issue.

	Type	Description
>	Unknown	Error while checking issue Type Scheme <b>Default Issue Type Scheme</b>
>	Unknown	Error while checking issue Type Scheme <b>Plane Issue Type Scheme</b>
>	Missing object	Role <b>Developers</b> refers to the missing user <b>yoda</b>
>	Unknown	Error while checking screen Scheme <b>Broken Screen Scheme</b>
>	Unknown	Error while checking screen Scheme <b>More Broken Scheme</b>
>	Unknown	Error while checking issue Type Screen Scheme <b>Simple issue type screen scheme</b>
>	Missing object	Workflow <b>Sales Workflow</b> refers to the missing group <b>Force</b>
∨	Missing object	Workflow <b>Sales Workflow</b> refers to the missing custom field <b>customfield_10201</b>

**Location:**

```

graph LR
    A[Sales Workflow (WORKFLOW)] --> B[Open (1) (STEP)]
    B --> C[Close (31) (TRANSITION)]
    C --> D[User Is In Group Custom Field (CONDITION)]
    D --> E[customfield_1020 (CUSTOM FIELD)]
          
```

**Solution:**  
Modify workflow **Sales Workflow** to point to an existing custom field or remove the reference to the missing custom field.

>	Missing object	Workflow <b>Sales Workflow</b> refers to the missing role <b>10110</b>
>	Missing object	Workflow <b>Sales Workflow</b> refers to the missing custom field <b>customfield_10100</b>
>	Missing object	Workflow <b>Sales Workflow</b> refers to the missing role <b>10110</b>
>	Missing object	Workflow <b>Sales Workflow</b> refers to the missing role <b>10110</b>
>	Missing object	Project <b>Sample1</b> refers to the missing user <b>yoda</b>
>	Missing object	Project <b>Sample1</b> refers to the missing group <b>Force</b>
>	Missing object	Project <b>Sample1</b> refers to the missing user <b>yoda</b>
>	Unknown	Error while checking project <b>Sample1</b> : java.lang.NullPointerException
>	Missing object	Project <b>Sample2</b> refers to the missing user <b>yoda</b>

Bug tracking and project tracking for software development powered by Atlassian JIRA (v6.2.3#6260-sha1:63ef1d6) · [About JIRA](#) · [Report a problem](#)

6. Declare **SUCCESS/FAILURE** based on the deployment result. If it is successful and the access to the users was limited, communicate and open the system for all users.
7. In case of **FAILURE** restore the snapshot on the production server, and start again. If the staging server is identical to the production, failures at this point aren't possible.

✔ [How to automate the promotion of Jira project configuration from test through staging to production Jira environment with Configuration Manager for Jira](#)

## Moving add-on data

Migrating the add-on data is one of the most common data portability issues we've encountered. In this section, we will explore various options for effectively moving the add-on data.

The data created and stored by each add-on can be categorized based on its usage and storage mechanism.

- **User-created data.** For example the hierarchical structures created by the Structure add-on include data created and consumed by users. It is not related anyhow to the project, system, or even add-on configuration. This type of data is not handled by Configuration Manager for Jira and the add-on should provide the export/import functionality.
- **Configuration data.** The add-on configuration could contribute to workflow post functions/conditions /validators, custom fields, dashboard gadgets, and other Jira objects, or could be private for the add-on and stored in the database (e.g. using [Active Objects](#).) Configuration Manager supports several [add-ons](#) out of the box, other add-ons can be easily made compatible if the add-on vendor implements SPI provided by Configuration Manager.

## Limitations & workarounds

There are certain **system limitations** that need to be considered during a configuration roll-out, including:

- **Add-ons data** Certain add-on custom field configuration, add-on workflow property configuration, and any other add-on data outside of Jira configuration objects might not be included.
- **Jira Service Desk** Jira Service Desk-specific configuration is not supported. It will be added in future releases of Configuration Manager.
- **Differences in configuration objects in 6.x to 7.x** This should be considered when the snapshot is created on a different major version of Jira than the target. 7.x has removed some permission types (e.g. the USE global permission is replaced by Application roles) and has introduced various other configuration objects, specifically permissions for permission schemes (e.g. manage sprints), security level type (application role), visibility permission (logged in users.)

Suggested **workarounds** regarding limitations and other known issues include:

- **Scripts** API level scripts can be developed to migrate any data that is not handled by Configuration Manager for Jira. Great choice for scripts development is using the [ScriptRunner](#) add-on.
- **Manual** If the amount of configuration is not extensive it can be manually added to the target Jira.
- **Professional Services** Botron's team of solution architects can develop a custom solution that migrates any type of data.

## Common issues

We have identified several common issues during deployments, which could negatively impact the data portability process. Some of the most common issues include:

- **Integrity Check** errors which prevent the snapshot creation/deployment. To preserve the integrity of the target/production server, Configuration Manager [Integrity check](#) is executed every time a snapshot is created or deployed. All critical errors reported should be resolved in order to continue. Detailed information can be obtained [here](#).
- **Differences in Jira versions** differences in the source-target Jira versions, specifically major versions.
- **Application permissions/licenses** for 7.x, e.g. deploying a software project when the user performing the deployment doesn't have permissions to create software projects; or the Jira Software application isn't installed/licensed.

- **Inconsistencies in user directories**if test/stage/production environments don't have the same user directory, user creation may be required if a new project is deployed and the project's lead doesn't exist in the target's user directories.
- **Proxy/firewall**this may be problematic for large instances when creating a snapshot or before the **Analysis** phase when deploying, otherwise there shouldn't be any issues with the deployment itself. This is worked around by increasing any timeout limits on the proxy servers and disabling the firewall blocking rules.

## Frequently Asked Questions

1. Is my entire configuration going to be rolled out to production automatically using Configuration Manager for Jira?  
*Yes. The entire configuration, captured in the configuration snapshot, will be rolled out. The supported configuration objects types which are included in the snapshots are listed [here](#).*
2. Can I roll out project configuration changes from a newer server instance to an older server instance?  
*It is strongly recommended that your production and staging server are on the same version. Configuration Manager allows deployment between different versions and warns the user about that. Snapshots created on newer versions may not work on old ones, as they might include new configuration objects which do not exist in the older versions for example in Jira 7.x there are new permissions which do not exist in 6.4.x.*
3. How long does it take to deploy the configuration snapshot?  
*Project configuration snapshots take in most cases from a few seconds to 1 minute. Large system snapshots which include hundreds of projects, thousands of filters, and hundreds of Agile boards could take more than 1 hour to deploy. An accurate estimate of the deployment time should be obtained during the staging.*
4. What if an error occurs during deployment? Does that break the target/production instance?  
*All changes deployed by Configuration Manager are executed in a single transaction. If an error occurs, the whole **transaction** is rolled back so the only time the server configuration is modified is when all the changes are deployed successfully.*
5. Can I automate the snapshot creation and deployment?  
*Yes. Configuration Manager provides a public **REST API** for this purpose.*
6. I can't create configuration snapshot due to Integrity check errors. Why is there such a limitation? How can I resolve this problem?  
*The **Integrity check errors** could indicate a serious problem. Other not critical errors are marked as warnings and they don't block the creation or deployment of snapshots. By not allowing a deployment of configuration with critical errors, Configuration Manager protects the production system from being modified with something that is not working.*
7. Can Configuration Manager handle custom-built add-ons?  
*Yes. Configuration Manager can be extended to handle any custom add-ons. Contact Botron's **services** team for more information.*

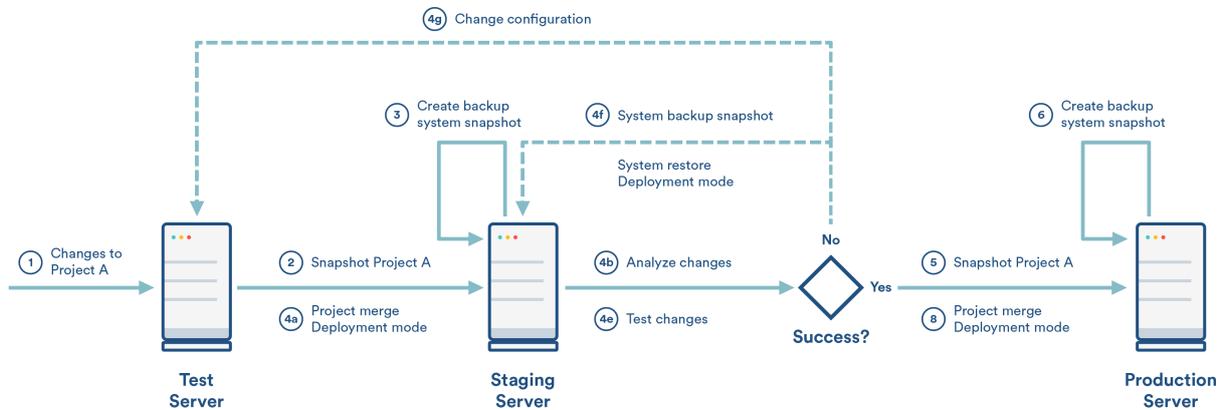
## Need help?

If you cannot resolve the problem yourself, you can contact us for assistance via the following communication channels:

- Send email to [support@botronsoft.com](mailto:support@botronsoft.com)
- For training or services specific information, contact [services@botronsoft.com](mailto:services@botronsoft.com)

# Three stages in detail

The procedure of staging any changes before they are made on the **production** server is a recommended IT best practice. The following article describes how to set up such environments: [Establishing staging server environments for Jira applications](#).



## Steps

This use case has two major phases: promotion from test to staging and promotion from staging to production.

### Test to staging

1. **Test:** Create all required changes to project A.
2. **Test:** Create a snapshot for project A.
3. **Staging:** (optional) Create a system snapshot for backup.
4. **Staging:** Deploy the snapshot.

- a. Use the [Project Merge](#) mode.
- b. Review the change and impact analysis in the [Analyze](#) step of the deployment. Now that **staging** is a close or identical replica of the **production**, during this analysis, you will see how the deployed configuration will impact the **production** server.
- c. Proceed with the deploy (the change and impact analysis shows that everything is OK.)
- d. Review the [Audit log](#) for any warnings.
- e. Test the deployed changes; the actual testing depends on the changes and may include creating issues, exercising issues workflow, etc. Declare **SUCCESS/FAILURE** if the deployment and testing are successful; proceed with promotion to **production**.
- f. In case of failure, locate the error and make changes in the **test** environment to address it.

**Note** do not make changes directly to the **staging** server; it needs to remain the same as **production**.

- g. In case of failure, restore at the **staging** server the system snapshot created at step 3 above.

### Staging to production

5. **Staging:** Create a snapshot of project A.
6. **Production:** (optional) Create a system snapshot for backup.
7. **Production:** (optional) [Limit the user access](#) to Jira. It is recommended that all changes to production servers are done during **maintenance windows** when the user access is limited.
8. **Production:** Deploy the snapshot.

- a. Use the [Project Merge](#) mode.

- b. Review the change and impact analysis in the [Analyze](#) step of the deployment.
- c. Now that the changes were staged at the **staging** instance, the change and impact analysis shown at this step should be identical with the one on staging. Those changes were tested and you can proceed with confidence.
- d. If the change and impact analysis is different than the one on **staging**, then both environments are different and this should be corrected.
- e. Proceed with the deployment (the change and impact analysis shows that everything is OK.)
- f. Review the [Audit log](#) for any warnings.
- g. Declare **SUCCESS/FAILURE** based on the deployment result. If it is successful and the user access was limited, open the system for all users.
- h. Only in case of failure restore at the **production** server the system snapshot created at step 6 above. Restart the process from the beginning. If the **staging** server is identical to the **production**, failures at this point aren't possible.



Note **do not** make changes directly to the **production** server

## Automation

Automation of the above steps can be accomplished by using the public [REST API](#).

Let us know what you think

[Feedback](#)

# Migrating Jira projects

 Tasks described on this page are not native to Jira. To complete them, you'll need to install a 3rd party add-on [Configuration Manager for Jira](#).

## Overview

This document describes best practices for moving projects between Jira server instances. The described migration processes are performed with the help of the [Configuration Manager for Jira](#) add-on. In addition to moving Jira projects, the described approach can be used for consolidating efforts such as merging multiple Jira instances.

- [Architecture Strategy Recommendations](#)
- [Planning](#)
- [Stages](#)
- [Limitations & Workarounds](#)
- [Common issues](#)
- [Frequently Asked Questions](#)
- [Need Help?](#)

## Definitions

For this document, we'll assume the following definitions:

- **Development** A free-for-all one or many environments where users can play with cutting-edge or risky changes.
- **Staging** A pre-production environment, where the systems administration team can establish exact procedures prior to rollout. The staging should be a clone or close replica of the production environment.
- **Production 1 (Source)**: Your live instance where the project(s) originate, expecting minimal downtime and well-tested changes.
- **Production 2 (Target)**: Your live instance where the project(s) should be moved, expecting minimal downtime and well-tested changes.
- **Configuration snapshots** are created using the [Configuration Manager for Jira](#) add-on and represent the state of your Jira [configuration objects](#) and their relations to each other at a given point in time. There are two types of configuration snapshots:
  - **Project snapshot** Contains the configuration of a number of selected projects (with their schemes, workflows, fields, etc.)
  - **System snapshot** Contains the entire configuration of a single Jira instance (projects, workflows, schemes, screens, etc.)

## Architecture Strategy Recommendations

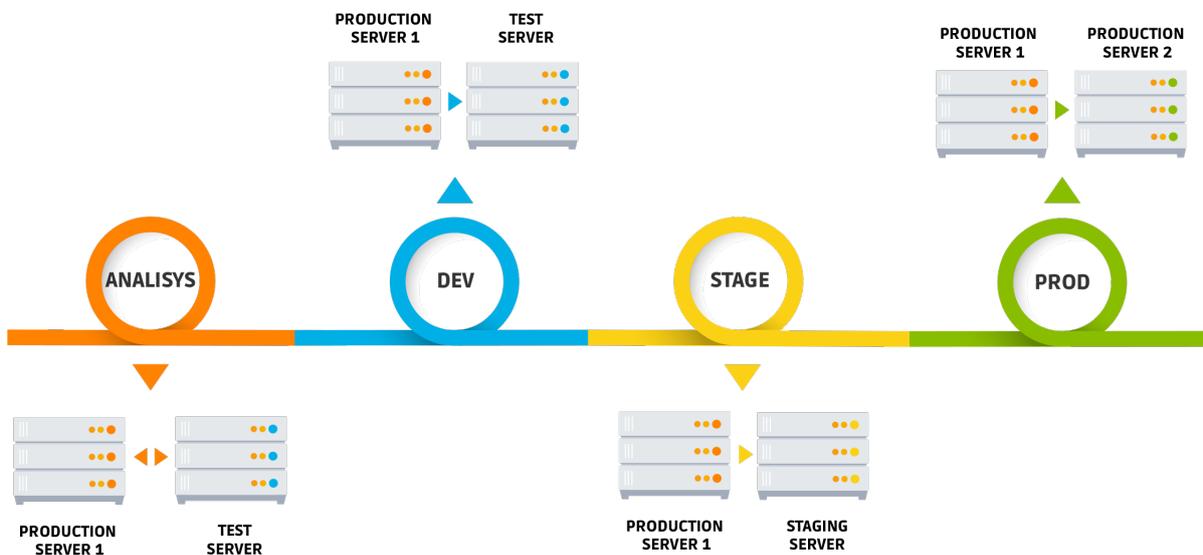
### Data

This guide presents how to run a transparent, end-to-end migration with no data loss. The majority of the [data objects](#) are handled out-of-the-box by the Configuration Manager for Jira. All exceptions and ways to handle them are listed in the [Limitations](#) sections of this guide.

### Process

The moving of projects between **Source** (Production 1) and **Target** (Production 2) instances will result in changes in your **Target** (Production 2) instance. We recommend that you follow the [Test -> Staging -> Production](#) procedure. Before the migration, a significant analysis should be performed. These activities are grouped under the Analysis phase in this guide. The major difference between the process of moving projects and rolling-out configuration changes is that in the first case scenario all user-generated data needs to be transferred, which requires a significant up-front analysis. The process of moving multiple projects between Jira instances is comprised of four major stages:

- Analysis - during this stage an analysis of the conflicts and gaps between the project configurations is performed; see [Application Migration Specification](#).
- Development - during this stage [Migration Procedure](#) document is developed and tested on the test server.
- Staging - during this stage the Migration Deployment procedure is staged and tested on the staging server.
- Production - during this stage the migration deployment procedure is executed in the official production environment.



## Planning

The following planning steps described in the [Test-Staging-Production](#) procedure use case should be completed before the migration:

- Prepare the staging environments & keep them synchronized
- Install Configuration Manager for Jira
- Track changes with change request tickets
- Plan disruptive & non-disruptive changes during maintenance windows
- Prepare communication strategy

Additionally, you should include the following planning steps:

- Prepare backup of both production servers
- Assess the condition of both, source and target, instances and fix any detected issues. System health assessment should be performed with [Jira Integrity checker](#) and Configuration Manager for Jira [Integrity Check](#).

## Stages

### Stage One: Analysis

Overview of the first stage:

**Jira Instances:** Production 1 (or a clone), Test Server (clone of Production 2)

**Goals:** Prepare Application Migration Specification (AMS) document

**Users:** Jira System Administrator, AD admin, DBA.

**Tools Required:** [Configuration Manager for Jira](#)

Below is the list of recommended analysis. Note that any changes that need to be made should be included in the AMS document:

### 1. Add-ons and Versions

Both Prod 1 and Test Server meet the following requirements:

- Running the same Jira version
- Have the same set of add-ons with matching versions

### 2. User Management Normalization

If both production servers are using the same external user directory, there should be a full match between the users and groups on both servers. If the match is not full, an analysis should be performed to identify gaps and conflicts.

#### Example

- i. **Gap:** On Prod1 Jane Smith has username *jsmith*, while on Prod 2 server her username is *jane.smith*
- ii. **Conflict:** Both servers have user John Smith with username *jsmith*, but the usernames correspond to two different people

Similar issues apply to user groups.

Typically, the list of conflict and potential matches should be prepared and then the business users should provide information how to resolve it.

#### Warning

If the user management is not normalized, a wide range of negative effects will take place. All fields where users or groups are used - Assignee, Reporter, etc. might contain wrong user or group. All configuration objects with users or groups could end up improperly configured - permissions, notifications, workflow customization with users and user group fields, filter and dashboard ownership, Agile board administrators, JQL filters with users or user group fields, issue security schemes etc.

#### How to resolve gaps and conflicts

The general approach for gaps and conflicts is to rename the username either on the Prod 1 or Prod 2 servers to ensure consistency. The renaming depends on the solution used for user management Crowd, AD, LDAP.

### 3. "As is" strategy

In most cases, the issues and project configuration are moved to the new production server "as is" without any modifications. Any modifications will need to be transformed to ensure consistency on both Jira instances. This process will ultimately add significant complexity and time to the migration and should be included in the AMS document. For this guide, an "As Is" strategy is used. For more information on the migration process with transformations, contact Botron's service team.

### ✔ Example

The Bug issue type might be part of different workflows on the two server instances. When you move a project with that issue, users have to choose between two options. In the "As is" case, the various projects use different workflows for the same issue type. In the *transformation* case, the two projects will use the same workflow, thus all the source data will be transformed to match the new workflow.

## 4. Configuration conflicts and gaps

When migrating projects their names or keys might already exist in the Prod 2 server, this conflict should be resolved by renaming them on the Prod 1 server. The same can happen for custom fields, resolution, priorities, workflow names and schemes. The full list of conflicts and the mappings of their resolutions should be included in the AMS document. The reverse scenario is possible too when two custom fields on Prod 1 and Prod 2 have different names but the same semantics. In this case, they ought to be merged during the migration.

### ✔ Tips

1. This analysis of these configuration conflicts and gaps can be done with the Configuration Manager for Jira change and impact analysis feature. Prior to the migration, the Configuration Manager for Jira enables you to perform a change and impact analysis and provides you with comprehensive details regarding the impact of the changes that will be applied to your Jira instance. By doing this analysis, you will be able to see the projects that will be affected by the change, identify any conflicts and gaps and resolve them. The list of introduced changes and their corresponding conflicts and gaps are grouped in tabs in the same order following that of a Jira menu.

2. This analysis is a great opportunity to optimize the usage of custom fields and reuse them between the two system instead of creating new ones. This will greatly improve the performance of the server.

## 5. Default Schemes

Any default schemes used on the Prod 1 server by the migrated projects should be changed. To change the default schemes:

- a. Copy the used default scheme.
- b. Rename the newly created scheme.
- c. Associate the project(s) with the new scheme.

## 6. Quality Strategy

A quality strategy should be developed and agreed upon by all stakeholders. The recommended quality strategy consists of test plans for each of the phases, and a post-migration remediation plan.

## 7. Test Plans

**Development phase** a set of tests should be executed to verify the following:

1. Filters - number of issues, issue ordering, column layout
2. Dashboards - layout (columns, rows), gadgets & gadget content
3. Agile
  - a. Board views (i.e. backlog, active sprints)
  - b. Issues, ranking, epics, versions, estimates, time tracking data aggregations
  - c. Quick filters
  - d. Sprints ordering & content
  - e. Reports
  - f. Project <-> Board associations
4. Issues
  - a. Agile data -sprints,epic links
  - b. Comments
  - c. History
  - d. Field values - 3-rd party custom fields, time tracking and estimates, other
  - e. Issue links

- f. Confluence links
- g. Bitbucket links (branches, commits)
- 5. Security
  - a. Projects - role memberships, access and operations for different roles
  - b. Boards, Filters, and Dashboards - ownership and operations
- 6. Operations - transition through workflow, create/deleted/edit issues, comments ,etc.

**Staging phase** - the tests from the development phase are executed again and a User Acceptance Test (UAT) is performed by the users of the projects that is being migrated.

**Production phase**, a subset of the development phase tests are executed plus a limited UAT test. Typically, there is a time limit on how long can these tests be performed in production. This time limit is aligned with the duration of the maintenance window during which the production phase takes place.

**Post Production Remediation Plan** - this plan covers the procedures designed to handle any post-migration issues that occurred during the production. It includes a SLA for response and resolution.

## Stage Two: Development

Overview of the second stage:

**Jira Instances:** Clone of Production 1, Test Server (clone of Production 2)

**Goals:** Prepare Migration Procedure document

**Users:** Jira System Administrator, AD admin, DBA.

**Tools Required:** [Configuration Manager for Jira](#)

Each of the migration tasks should be properly documented and put in an ordered list. It should include any input data and enough details so that it can be repeated consistently.

The recommended approach for documenting is a structure of tasks and steps to accomplish each task.

1. For each of the data or configuration changes included in the AMS document, execute the required steps.
2. **Create project snapshot** that includes the projects that are being migrated from **Prod 1 server**. Include all related to the projects Agile boards, filters, dashboards and issues

 Issues support is still in Beta mode. We're making our best to provide official support for this with Configuration Manager 5.0.

**Create Configuration Snapshot**

Progress: Select | Filters | Boards | Dashboards | Preview | Create

### Preview Snapshot

Details

Name:	Type:	Description:
asd	Project (HSP)	N/A

▼ Filters (3 of 5)

Name	Owner	Exclude
My Approvals	Administrator	Exclude
My Milestones	Administrator	Exclude
My Tasks	Administrator	Exclude

▼ Boards (2 of 2)

Name	Owner	Exclude
TSP board	admin	Exclude
SOM board	admin	Exclude

▼ Dashboards (2 of 2)

Name	Owner	Exclude
Dashboard 1	Administrator	Exclude
System Dashboard		Exclude

Buttons: Back | **Create** | Cancel

3. **Download the configuration snapshot.** If your Jira configurations are being tracked via tickets, the snapshot should be attached to the appropriate tickets.
4. **Deploy configuration snapshot to Test Server** and validate proposed configuration changes.

**Deploy Configuration Snapshot**

Progress: Select | Analyze | Migrate | Deploy

Analyze Configuration Changes

ITP: Simple Issue Tracking Workflow

Changes:

- Property statuses removed from condition "Condition on subtasks":
  - At least 0 and no more than 1000 subtasks with the following characteristics:
    - Issue types: License, Dev Task or Doc Task.
    - Statuses: Failed, Review or Resolved.
    - Resolutions: Cannot Reproduce, Done or Won't Fix.
  - About the rest of subtasks:
    - Unselected issue types are not allowed.
    - Unselected statuses are not allowed.
    - Unselected resolutions are not allowed.
- Property statuses added to condition "Condition on subtasks":
  - At least 0 and no more than 1000 subtasks with the following characteristics:
    - Issue types: License, Dev Task or Doc Task.
    - Statuses: Failed, Review or Resolved.
    - Resolutions: Cannot Reproduce, Done or Won't Fix.
  - About the rest of subtasks:
    - Unselected issue types are not allowed.
    - Unselected statuses are not allowed.
    - Unselected resolutions are not allowed.

Buttons: Back | Next | Cancel

5. It is crucially important to make sure the proposed configuration [changes and impact](#) match the AMS.
  - a. If they don't match AMS, the changes made in step 2 need to be corrected. Restore your backup on the clone of Production 1 and the Test Server and resume from step 1.
  - b. If they match AMS, continue.
6. Update the ticket with the required deployment time for the snapshot deployment.
7. Execute the test plan for test stage
  - a. If the tests fail, they have to be resolved. Restore your backup on the clone of Production 1 and the Test Server and resume from step 1.
  - b. If the tests pass, attach the Migration procedure to the migration ticket and proceed to Stage 3.

The steps described in this stage should be repeated several times to ensure that everything runs smoothly and both the proposed changes and the associated tests are successful.

## Stage Three: Staging/QA

Overview of the third stage:

**Jira Instance:** Production 1 (or a clone), Staging Server (close replica of Production 2)

**Goals:** This is the grand rehearsal of the migration and has two major goals:

1. Verify the migration procedure and conduct an UAT.
2. Get an accurate time estimate for the duration of the migration.

**Users:** Jira System Administrator, AD admin, DBA, business users to conduct UAT.

**Tools Required:** [Configuration Manager for Jira](#)

The **steps** of this stage are as follows:

1. Execute each of the tasks from the migration procedure without any modifications.
  2. Estimate the required downtime.
  3. Perform development stage tests.
  4. Perform the UAT tests.
- If both the development phase test and the UAT are successful, declare successful staging and proceed to production.
  - In the case of failure, depending on the type of issue(s), either change the deployment procedure and repeat the staging or go back to development phase to correct the issue(s).

## Stage Four: Production Deployment

**Jira instance:** Production 1, Production 2

**Goals:** Finish the migration in the official Production 2

**Users:** Jira System Administrator, AD admin, DBA.

**Tools Required:** [Configuration Manager for Jira](#)

The fourth and final stage of the migration process is, for the most part, identical to the staging stage. The major difference being that you'd be working with the actual Production 1 and 2 servers, not their clones.

The **steps** of this stage are as follows:

1. Create full backups of both production servers.
2. Restrict the user access to Production 2 server.
3. Execute each of the tasks from the migration procedure as they are captured without any modifications.
4. Execute development stage tests.
5. Execute the UAT tests.
6. If both the development phase test and the UAT are successful, declare successful migration and open the access for the users.
7. If not successful, depending on the type of issues found there are two options:
  - a. Apply changes/fixes to Production 2 server and rerun the UAT tests.

- b. Restore the backups and reschedule the production deployment until the issues are resolved.

## Limitations & Workarounds

There are certain **system limitations** that need to be considered:

- **Add-ons data:** Certain add-on custom field configuration, add-on workflow property configuration any other add-on data outside of Jira configuration objects might not be included. A custom solution needs to be developed.
- **Jira Service Desk:** Jira Service Desk-specific configuration is not supported OOTB. It will be added in future releases of Configuration Manager.

Suggested **workarounds** regarding limitations and other known issues include:

- **Scripts** - API level scripts can be developed to migrate any data not handled by Configuration Manager for Jira. Great choice for scripts development is using the [ScriptRunner](#) add-on.
- **Manual** - If the amount of configuration is not extensive it can be manually added to the target Jira.
- **Professional Services** - Botron's team of solution architects can develop a custom solution that migrates any type of data.

## Common issues

We have identified several common issues during deployments, which could negatively impact the data portability process. Some of the more common issues include:

- **Integrity Check** errors which prevent the snapshot creation/deployment. In order to preserve the integrity of the target/production server. Configuration Manager [Integrity check](#) is executed every time a snapshot is created or deployed. All critical errors reported should be resolved in order to continue. Details information can be obtained [here](#).
- **Differences in Jira versions:** differences in source-target Jira versions, specifically major versions.
- **Application permissions/licenses:** for 7.x, e.g. deploying a software project when the user performing the deployment doesn't have permissions to create software projects; or the Jira Software application isn't installed/licensed
- **Proxy/firewalls:** this may be problematic for large instances when creating a snapshot or before the [Analysis](#) phase when deploying, otherwise there shouldn't be any issues with the deployment itself. This is worked around by increasing any timeout limits on the proxy servers and disabling the firewall blocking rules.

## Frequently Asked Questions

1. Is my entire configuration and all the issues are going to be moved to production automatically using Configuration Manager for Jira?  
**Yes. The entire configuration, captured in the configuration snapshot, will be rolled-out. All the issues to the selected projects as well. Some add-ons configuration and data which are not supported by Configuration Manager for Jira won't be moved. The supported objects are listed [here](#).**
2. Can I move projects to a server with a newer server instance to an older server instance?  
**It is strongly recommended that both production servers are on the same version. The opposite adds unnecessary complexity.**
3. How long does it take to move multiple projects?  
**Depending on the user management normalization, the amount of data and the other analysis. It can take anywhere from several hours to few weeks.**
4. Can custom add-ons data be moved using Configuration Manager?  
**Yes. Configuration Manager can be extended to handle any custom add-ons. Contract Botron's [services](#) team for more information.**

## Need Help?

If you cannot resolve the problem yourself, you can contact us for assistance via the following communication channels

- Send email to [support@botronsoft.com](mailto:support@botronsoft.com)
- For training or services specific information contact at [services@botronsoft.com](mailto:services@botronsoft.com)

# Consolidating multiple Jira instances

The process of consolidating multiple Jira instances is exactly the same as the migration process described in Migrating Jira projects.

For the complete documentation on Migrating Jira projects, see [here](#).

While the end result is different, the approach, tools, planning and overall staging procedure is the same. To consolidate multiple Jira server into one, you need to use the Configuration Manager for Jira to merge system and project level configuration data, user-generated data, 3rd party add-on configuration and other related information. To ensure consistency in your merger, a comprehensive analysis, testing and staging of the consolidation efforts should be performed prior to rolling out the changes of your final merger. Any errors, gaps, conflicts and merge-specific issues that you identify should be resolved during the analysis phase of the consolidation process.

If you face any challenges during your consolidation efforts or cannot resolve the problem on your own, feel free contact us for assistance through the following communication channels:

- Send email to [support@botronsoft.com](mailto:support@botronsoft.com)
- For training or services specific information contact [atservices@botronsoft.com](mailto:atservices@botronsoft.com)

# Configuring Jira application emails

To get the most out of your Jira applications, you can configure them to send email notifications when an event occurs, and to perform tasks on the receipt of an email.

Search the topics in 'Configuring Jira application emails':

## Sending emails

Your Jira applications can send emails when an event occurs, such as an issue is created or completed, or when the issue is updated. These are called email notifications.

[Learn more](#) about configuring the email notifications and customizing the content.

## Receiving emails

Your Jira applications can be configured to perform tasks when they receive emails. You can choose to allow your applications to create new issues when an email is received, or update an existing issue with a comment.

[Learn more](#) about configuring the incoming mail options and creating mail handlers.

# Configuring email notifications

Jira can send email notifications to users when significant events occur. For example, you can get notified when an issue is created or completed, or when any of the issue fields is updated.

 For all of the following procedures, you must be logged in as a user with the [Jira Administrators global permission](#).

## Skip to

- [Enabling notifications](#)
- [How notifications work](#)
- [Changing the notification frequency](#)
- [Customizing email content](#)
- [Configuring a project's email address](#)
- [Email recipients](#)
- [Email format: HTML or text](#)
- [Troubleshooting notifications](#)

## Enabling notifications

To start the notifications flowing, you'll need to set up an **SMTP mail server**, and **create a notification scheme** in which you choose the notifications you'd like to get.

1. [Configure Jira's SMTP mail server](#)
2. [Create a notification scheme](#)

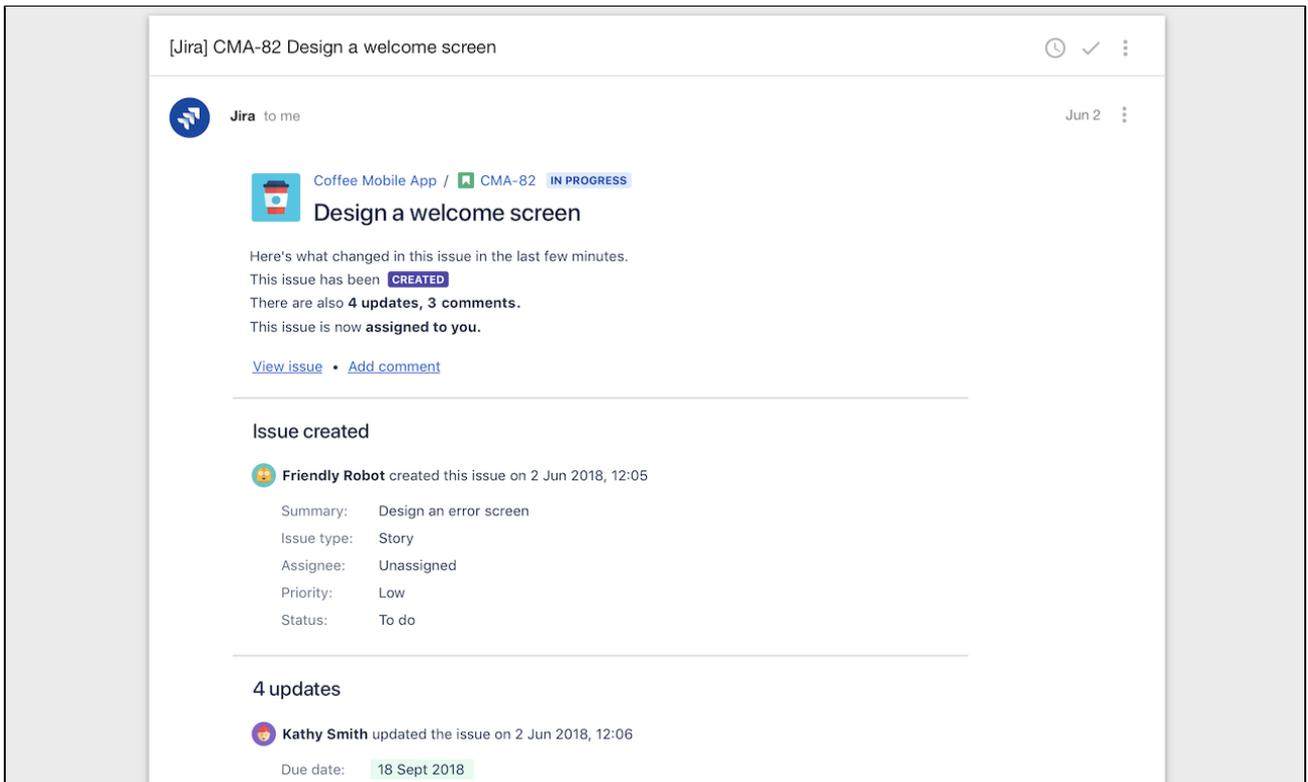
You can also customize your [email content](#), and configure the [email address](#) from which notifications are sent.

## How notifications work

Notifications for events that occur close together are grouped and sent in a summary email, just to avoid too many emails cluttering your mailbox. We're calling these notifications *batched notifications*, as opposed to *separate notifications* (where every notification is sent in a separate email) that were the default notifications in earlier Jira versions.

 Settings related to batched notifications are in **Administration**  **> System > Batching email notifications**.

An email with batched notifications looks like the one below. As you can see, it groups all events from the last few minutes and presents them as a summary.



### Issue updates

When it comes to notifications about issue updates, you will receive a separate email for every issue. The email contains events that occurred in an issue in the specified time frame (for example, the last hour), and includes the following updates:

- Details about the issue (if it was just created)
- Changes to any of the issue fields (including custom fields)
- Comments
- Work logs
- Attachments
- Mentions

You can [change the frequency](#) of batching these notifications. Depending on your choice, you'll receive an email with events from the last few minutes or an hour.

### Mentions

When you're mentioned in an issue, it usually means that somebody needs your immediate attention. To avoid sending too many emails, a mention will be included in the email with other issue updates, but it will trigger this email to be sent as soon as possible (regardless of the frequency you've chosen).

### Customer notifications in Jira Service Management

Jira Service Management has an additional set of notifications, which we call customer notifications. These notifications are designed to keep the customers informed about progress on particular tickets without disclosing information about internal processes, and usually include responses to customers, customer-visible changes, approvals, or requests being shared with individuals or organizations. You can read more about them in [Managing Jira Service Management notifications](#).

These notifications are not batched, and you can't choose the frequency of sending them every notification is sent in a separate email after an event occurs.

### Advanced: Adding more custom fields

Updates to custom fields in your issues are included in the notifications, just like regular fields. However, if you're an advanced Jira user, you can display additional custom fields in your emails. Such custom fields can be added to emails subject, header, or any other location. Some of our users do this to describe their notifications more precisely, for example by adding security levels. For more info, see [Adding custom fields to emails](#).

## Changing the notification frequency

Every team works in a different way, so we're giving you a way to tailor the notifications to your needs. If you like to always stay on top of things, you can receive batched notifications every few minutes (2 minutes being the minimum), and if you'd rather just read a summary every now and then, you can choose to receive a summary of what happened in the last hour.

To change the frequency:

1. Go to **Administration** (⚙️) > **System** > **Batching email notifications** (you can also search for it by pressing 'gg').
2. Choose the frequency that works for you, and click **Save**. These settings will be applied to all users.

### Disabling batched notifications

If you absolutely love to know what's going on in your projects, you can also disable batched notifications and get notified about every event (for example, change of status or any of the fields) in a separate email. You can choose to do this if you simply got used to separate notifications that were the default notifications before Jira 8.4.

To disable batched notifications:

1. Go to **Administration** (⚙️) > **System** > **Batching email notifications** (you can also search for it by pressing 'gg').
2. Select **Disabled**, and click **Save**. These settings will be applied to all users.

## Customizing email content

Email notifications are based on Apache's [Velocity](#) templates. Whether it's adding some info to the header or footer, or shuffling around the contents of your emails, you can make a number of customizations to the templates and tailor the emails to your needs.

For more info, see [Customizing email content](#).

## Configuring a project's email address

You can configure an email address for your project, which will work as the address that notifications are sent from - the sender address. This email address will also serve as the reply address for responses.

 This setting is specific to the selected project, and won't affect the configuration of other projects, or the default sender address of your SMTP mail server.

To configure a project's email address:

1. Choose **Administration** (⚙️) > **Projects**, and select a project.
2. Locate the **Notifications** section, and click the pen icon next to the email address.



## Notifications

Notifications allow Jira to send email notifications to specified people regarding particular events in your project. They'll receive a separate notification for each event.

Scheme: [Space Notifications](#)

Email:  [jira@atlassian.com](mailto:jira@atlassian.com)

3. Enter a valid email address, and click **Update**.

Notifications about issues from this project will now be sent from the new email address. If you ever wanted to change it back to the default address from your SMTP server, complete the same steps but leave the email address blank.

## Email recipients

For each notification, Jira will only send the first encountered email intended for a recipient. Hence, in the case where a user is included in two or more recipient lists (e.g. the **Project Lead** and current reporter) for one event notification, the user will only receive the first encountered email notification. Jira will log the fact that this user was on multiple recipient lists.

Jira can send email notifications to users when significant events occur, (e.g. creation of an issue; completion of an issue).

**Note:** For all of the following procedures, you must be logged in as a user with the **Jira Administrators** [global permission](#).

 Jira's default setting is to not notify users of their own changes. This can be changed on a per user basis via their profile preferences.

## Email format: HTML or text

Each Jira user can specify in their profile preferences, whether to send outgoing emails in either text or HTML format. **Jira Administrators** can specify a default email format by going to **Administration**  **> User Management > User Preferences**.

The HTML email format can accommodate internationalized words in the 'Issue Details' section. However, due to Internet Security Settings, which prevent images from being automatically downloaded, the HTML email messages may not be correctly formatted. For example, the summary column on the left may appear too wide. You can correct the formatting by accepting to download these images (this option should be available in the email).

## Troubleshooting notifications

### Using the Jira admin helper

The Jira admin helper can help you diagnose why a user isn't receiving email notifications when they should be, or why a user is receiving email notifications when they shouldn't be. This tool is only available to Jira administrators.

To diagnose why a user is or is not receiving notifications for an issue:

1. View the issue in Jira.
2. Click **Admin > Notification Helper**.
3. Enter the username of the user.
4. Click **Submit**.



You can also access the Notifications Helper by going to **Administration** (⚙️) > **System** > **Notification Helper**.

# Configuring an SMTP mail server to send notifications

To enable Jira to send [notifications](#) about various events, you need to first configure an SMTP mail server in Jira .

**Note:** For all of the following procedures, you must be logged in as a user with the **Jira System Administrators** global permission. **Jira Administrators** can enable or disable outgoing mail, but not configure SMTP mail servers.

## On this page:

- [Define or edit the SMTP mail server](#)
- [Specify a host name or JNDI location for your SMTP mail server](#)
- [Configuring a JNDI location](#)
- [Troubleshooting](#)

## Define or edit the SMTP mail server

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. Select **Mail > Outgoing Mail** to open the SMTP Mail Server page.
  - If no SMTP mail server has been defined, then a **Configure new SMTP mail server** button will be shown on the page. If one has already been defined, then the SMTP mail server's details will be shown on the page, along with a set of operation links at the right.
3. Click either the **Configure new SMTP mail server** button to define a new SMTP mail server, or the **Edit** link at the right to edit the existing SMTP mail server, which will open the **Add/Update SMTP Mail Server** page.
4. Complete the top section of this page as follows:

<b>Name</b>	Specify an arbitrary name to identify this SMTP mail server configuration.
<b>Description</b>	(Optional) Specify an arbitrary description that describes the SMTP mail server. This description appears below the <b>Name</b> of the SMTP mail server on the <b>SMTP Mail Server</b> configuration page.
<b>From address</b>	Specify the email address used in the 'sender address' (or 'from') field of notification messages sent by Jira, unless <a href="#">overridden in a project configuration</a> . <ul style="list-style-type: none"><li>• Only specify an email address for this field (e.g. <code>jira@example-company.com</code>). Jira will use this value to construct the full 'from' header based on the current user ("Joe Bloggs (Jira) &lt;<code>jira@example-company.com</code>&gt;").</li></ul> To change the 'from' header, go to <b>Administration &gt; System &gt; General Configuration</b> and (under <b>Settings</b> ), edit the <b>Email from</b> field.
<b>Email prefix</b>	Specify the subject of emails sent from this server will use this string as a prefix. This is useful for your users so that they can filter email notifications from Jira based on this prefix.

*Screenshot: Add (or Update) SMTP Mail Server*

## Add SMTP Mail Server ?

Use this page to add a new SMTP mail server. This server will be used to send all outgoing mail from JIRA.

Name \*   
The name of this server within JIRA.

Description

From address \*   
The default address this server will use to send emails from.

Email prefix \*   
This prefix will be prepended to all outgoing email subjects.

**Server Details**  
Enter *either* the host name of your SMTP server or the JNDI location of a `javax.mail.Session` object to use.

**SMTP Host**

Service Provider

Protocol

Host Name \*   
The SMTP host name of your mail server.

SMTP Port   
Optional - SMTP port number to use. Leave blank for default (defaults: SMTP - 25, SMTPS - 465).

Timeout   
Timeout in milliseconds - 0 or negative values indicate infinite timeout. Leave blank for default (10000 mSecs).

TLS.   
Optional - the mail server requires the use of TLS security.

Username   
Optional - if you use authenticated SMTP to send email, enter your username.

Password   
Optional - as above, enter your password if you use authenticated SMTP.

or

**JNDI Location**

JNDI Location   
The JNDI location of a `javax.mail.Session` object, setup by your application server.

## Specify a host name or JNDI location for your SMTP mail server

The second part of the **Add/Update SMTP Mail Server** page specifies the **Server Details** of the SMTP mail server to which Jira will send mail. There are two ways you can do this. Either:

- specify the **SMTP host** details of your SMTP mail server;
- or:
- specify the **JNDI location** of a `javax.mail.Session` object that is, use JNDI to look up an SMTP mail server that you have preconfigured in your application server. This has the following advantages:
  - **Better security:** the mail details are not available to Jira administrators through the Jira administration interface and are not stored in Jira backup files.
  - **More SMTP options:** for instance, you could switch to RSET instead of NOOP for testing connections by setting the `mail.smtp.userSet` property.
  - **Centralized management:** mail details are configured in the same place as database details and may be configured through your application server administration tools.

## Specify the SMTP host details

Most people configure Jira's SMTP mail server by specifying the SMTP host details of this mail server directly in Jira.

1. In the **SMTP host** section of the **Add/Update SMTP Mail Server** page ([above](#)), complete the following form fields:

<b>Service Provider</b> (not available when updating an existing SMTP mail server)	Choose between using your own SMTP mail server (i.e. <b>Custom</b> ), or either Gmail (i.e. <b>Google Apps Mail / Gmail</b> ) or Yahoo! (i.e. <b>Yahoo! Mail Plus</b> ) as the service provider for your SMTP mail server. <b>i</b> If you choose either Gmail or Yahoo! options and then switch back to <b>Custom</b> , some of the key fields in this section will automatically be populated with the relevant SMTP mail server settings for these service providers.
<b>Protocol</b>	Choose between whether your SMTP mail server is a standard (i.e. <b>SMTP</b> ) or a secure (i.e. <b>SECURE_SMTP</b> ) one.
<b>Host Name</b>	Specify the hostname or IP address of your SMTP mail server. Eg. <code>smtp.yourcompany.com</code>
<b>SMTP Port</b>	<i>(Optional)</i> The SMTP port number, usually 25 for SMTP or 465 for SMTPS, either of which are assumed if this field is left blank.
<b>Timeout</b>	<i>(Optional)</i> Specify the timeout period in milliseconds, which is treated as 10000 if this field is left blank. Specifying 0 or a negative value here will result in Jira waiting indefinitely for the SMTP server to respond.
<b>TLS</b>	<i>(Optional)</i> Select this checkbox if your SMTP host uses the Transport Layer Security (TLS) protocol.
<b>Username</b>	<i>(Optional)</i> If your SMTP host requires authentication, specify the username of these authentication credentials here. (Most company servers require authentication to relay mail to non-local users.)
<b>Password</b>	<i>(Optional)</i> Again, if your SMTP host requires authentication, specify the password associated with the username you specified above. <b>i</b> When editing an existing SMTP mail server, select the <b>Change Password</b> checkbox to access and change this field.

**⚠ Please note:**

- If your server's `startup script` uses the `-Dmail` system properties (e.g. `mail.smtp.host` or `mail.smtp.port`), they will override the settings that you specify in the above form. Additionally, if necessary you can manually specify the host name that Jira reports itself as to the SMTP server by setting `-Dmail.smtp.localhost`
  - The SMTP must support the multipart content type. Without this mails will not be able to send.
2. *(Optional)* Click the **Test Connection** button to check that Jira can communicate with the SMTP mail server you just configured.
  3. Click the **Add** (or **Update**) button to save Jira's SMTP mail server configuration.

**Specify a 'JNDI Location'**

As an alternative to specifying SMTP host details directly in Jira, you can configure them in your application server, and then look up a preconfigured mail session via JNDI.

In the **JNDI Location** section of the **Add/Update SMTP Mail Server** page ([above](#)), specify the location of a `javax.mail.Session` object to use when sending email, in the **JNDI Location** field. This will begin with the prefix `java:comp/env/`

**Configuring a JNDI location**

The **JNDI Location** that you specify in Jira will depend on Jira's application server and configuration. JNDI locations are typically configured in the application server that runs Jira. Hence, Jira will need to be restarted after configuring a JNDI location for that configuration to be available in Jira.

For example, in Tomcat 6 (the application server bundled with '[recommended](#)' [distributions of Jira](#)), your **JNDI Location** would be `java:comp/env/mail/JiraMailServer` and you would add the following section to the `conf/server.xml` of your [Jira application installation directory](#), inside the `<Context/>` node:

```
<Context path="" docBase="${catalina.home}/atlassian-jira" reloadable="false">
...
  <Resource name="mail/JiraMailServer"
    auth="Container"
    type="javax.mail.Session"
    mail.smtp.host="mail.yourcompany.com"
    mail.smtp.port="25"
    mail.transport.protocol="smtp"
    mail.smtp.auth="true"
    mail.smtp.user="jirauser"
    password="mypassword"
  />
...
</Context>
```

Or if you do not require authentication (e.g. if you are sending via localhost, or only internally within the company):

```
<Context path="" docBase="${catalina.home}/atlassian-jira" reloadable="false">
...
  <Resource name="mail/JiraMailServer"
    auth="Container"
    type="javax.mail.Session"
    mail.smtp.host="localhost"
    mail.smtp.port="25"
    mail.transport.protocol="smtp"
  />
...
</Context>
```

If you happen to be running Jira on an application server other than Apache Tomcat (which is [not a supported Jira configuration](#)), a similar methodology for configuring a JNDI location to your SMTP mail server should apply to that application server.

If you have problems connecting, add `mail.debug="true"` parameter to the `<Resource/>` element (above), which will let you see SMTP-level 'debugging' details when testing the connection.

### Move the JavaMail Classes

You will also need to ensure that the JavaMail classes (typically in JAR library files) are present in your application server's classpath and that these do not conflict with Jira's JAR library files. This is necessary because the application server itself (not Jira) is establishing the SMTP connection and as such, the application server can not see the JAR library files in Jira's classloader.

Some operating systems may bundle the JavaMail classes with application servers (e.g. **Tomcat in Red Hat Enterprise Linux**). This may conflict with Jira's copy of the JavaMail classes, resulting in errors like:

```
java.lang.NoClassDefFoundError: javax/mail/Authenticator
```

or:

```
java.lang.IllegalArgumentException: Mail server at location [java:comp/env/mail/JiraMailServer] is not
of required type javax.mail.Session.
```

Lighter application servers such as Apache Tomcat (including the one incorporated into the 'recommended' distributions of Jira), do not always come with JavaMail.

To prevent any conflicts, check your application server's `lib/` directory:

- If the application server already contains `mail-1.4.1.jar` and `activation-1.1.1.jar`, then just **remove** `mail-1.4.1.jar` and `activation-1.1.1.jar` from the `<jira-application-dir>/WEB-INF/lib/` subdirectory of the [Jira application installation directory](#).

- If the application server does not contain `mail-1.4.1.jar` and `activation-1.1.1.jar`, then **move** the `mail-1.4.1.jar` and `activation-1.1.1.jar` from the `<jira-application-dir>/WEB-INF/lib/` subdirectory of the [Jira application installation directory](#) into the `lib/` subdirectory of the Jira installation directory (for 'recommended' distributions of Jira) or the `lib/` subdirectory of the application server running Jira.

### SMTP over SSL

You can encrypt email communications between Jira and your mail server via SSL, provided your mail server supports SSL.

Firstly, you will need to **import the SMTP server certificate** into a Java keystore. The process is described on the [Configuring an SSL connection to Active Directory](#) page.

**⚠ Important Note:** Without importing the certificate, Jira will not be able to communicate with your mail server.

Secondly, edit your mail server connection properties and specify `starttls` and `SSLSocketFactory`. From `{Jira_INSTALL}/conf/server.xml` (this example uses Gmail's server):

```
<Resource name="mail/GmailSmtpServer"
  auth="Container"
  type="javax.mail.Session"
  mail.smtp.host="smtp.gmail.com"
  mail.smtp.port="465"
  mail.smtp.auth="true"
  mail.smtp.user="myusername@gmail.com"
  password="mypassword"
  mail.smtp.starttls.enable="true"
  mail.smtp.socketFactory.class="javax.net.ssl.SSLSocketFactory"
/>
```

## Troubleshooting

A useful tip for debugging mail-related problems in Jira is to set the `-Dmail.debug=true` property on startup. This will cause protocol-level details of Jira's email interactions to be logged. Additionally, [turning up JIRA's log level](#) will show when the service is running and how mails are processed.

### Common Problems

- If Jira does not appear to be creating or sending emails or creating issues and comments from email, your Jira installation could be experiencing **OutOfMemory errors**. Please check your log files for OutOfMemory errors. If there are OutOfMemory errors, please restart Jira and [investigate the errors](#).
- If you find some incoming emails simply disappear, check that you have not accidentally **started a second copy of Jira** (eg. in a staging environment) which is downloading and deleting email messages. See the [Restoring data](#) page for flags you should set to prevent mail being processed.
- If you receive 'Mail Relay' errors, make sure you have specified the **Username** and **Password** in the **SMTP Host** section of Jira's **SMTP Mail Server** configuration page.

### Getting Help

If you cannot resolve a problem yourself, please [create a support case](#) in the 'Jira' project and we will assist.

# Customizing email content

Jira generates emails in reaction to events using a templating engine [Apaches Velocity](#). This is a relatively easy to use templating language that can pull apart java objects in useful ways. Whether its adding some info to the header or footer, or shuffling around the contents of your emails, you can make a number of customizations and tailor the emails to your needs.

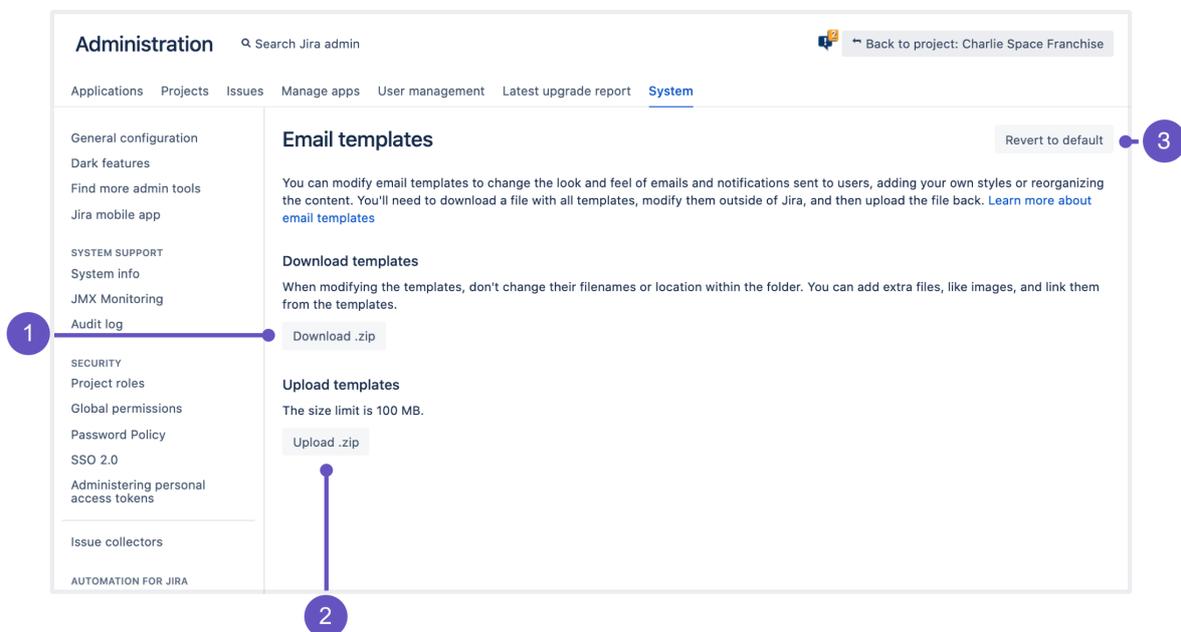
## This page is for Jira Data Center

If youre using Jira Server, youll need to retrieve your templates in a different way than described here. For more info, see [Customizing email content in Jira Server](#).

## Accessing your email templates

To access your email templates:

- Go to **Administration > System > Email templates**. You need to be a Jira System Administrator to complete any actions on this page.



1. **Download:** Download a ZIP archive with all available templates and edit them locally.
2. **Upload:** Once youre done editing, upload the archive back to Jira. The archive needs to have the same file structure as the original. See [Required file structure](#) below.
3. **Revert to default:** If there are problems with your uploaded templates, revert them to default ones. Jira will copy the default templates from its resources to your Jira shared home directory.

## Editing your email templates

Editing email templates allows you to change the look and feel of your email notifications. This comes in handy if youre looking to add your company branding or reorganize the content. When editing the template, youll be using the Velocity templating language, and some knowledge of it will be essential. To see the list of all templates and their description, check:

- [List of available email templates](#)

## Uploading the templates

You'll need to archive the templates into a ZIP archive and upload it back to Jira. Here are some things you'll need:

### Required file structure

When editing the templates, you shouldn't change their filenames or location within the folder, as Jira will validate the file structure. You can add extra files to the archive, like graphics, and link them from your templates.

Jira only checks if the required templates exist. The syntax or any errors in your templates won't be validated.

The following file structure will be accepted. Alternatively, you can omit the templates folder:

```
- templates
  - email
  - email-batch
```

### Applying the changes

After uploading the templates, they won't be applied right away. Jira will first validate the file structure, and then ask you to apply the changes if everything looks right. Once you do it, the new templates will be moved to <Jira-shared-home>/data/templates, overwriting whatever exists there.

### Reverting templates to default

If something isn't right with your modified templates, you can bring the default ones by clicking **Revert to default**. Here's what happens when you do it:

1. Jira copies the default templates from its resources to the Jira shared home directory.
2. The default templates overwrite any existing templates in <Jira-shared-home>/data/templates.
3. Jira clears the template cache so the default templates are applied right away.

### Good to know

Here is some other information that you might find useful.

### Templates: Default vs. current

#### Current templates

Current templates are templates that Jira is using for email notifications. They are stored in your Jira shared home directory and are updated every time you upload a new ZIP archive and apply the changes.

- **Separate issue notifications (and other events):** <Jira-shared-home>/data/templates/email
- **Batched issue notifications:** <Jira-shared-home>/data/templates/email-batch

#### Default templates

Default templates are backup templates that are stored in the Jira resources, either in the installation directory (separate issue notifications and other events) or in one of the plugins (batched issue notifications). It's what you would edit to customize your emails before Jira 8.14. These templates work as backups and will only be used in these cases:

- **Missing files:** One of the templates is missing in your Jira shared home directory. For this template, Jira will fall back to the default one.
- **Revert to default:** You chose to revert your templates to default. Default templates will be copied to your shared home directory, overwriting the current templates.

✔ Because of how default templates work, we recommend against modifying them so Jira always has something to revert or fall back to.

### Changes retained on upgrade

Since email templates are stored in your Jira shared home directory, any changes you make will be retained on upgrade. Also, you don't have to restart Jira after editing the templates.

### Caching email templates

All templates in the Jira shared home directory are cached. Thanks to that, Jira can use these templates even if the shared directory isn't accessible. We don't cache information about missing files. If one of the templates is missing from the shared directory, Jira will try to read it every time and will fall back to the default template.

# Templates: Batched issue notifications and other events

The archive that you downloaded from Jira contains email templates from all available configuration options. Its easy to get lost in the files and their dependencies, especially that not all of them apply to your configuration. Use this page to understand the templates and the notifications they control.

## This page is for default Jira notifications

This page is for default Jira notifications that are using batched issue updates. If youre using separate issue notifications, see [Separate issue notifications](#).

[Not sure which notifications youre using?](#)

## Definitions

Were referring to templates and includes around this page. Since all of them are Velocity templates, heres the best way to understand them:

- **Templates:** Main templates used for notifications. Usually, a template like that is a notification that youre getting. Youd modify them if youd like to customize the looks of a specific notification rather than an item that is shared with all notifications, like footer.
- **Includes:** Includes arent used on their own, but are always referenced in main templates. Youd modify them if youd like to customize the looks of an item shared with all notifications rather than the notification itself. For example, changing footers or headers.

## Issue notifications

### Templates

These templates control notifications about issues (summary emails).

Directory: `templates/email-batch/html`

Template	Description	Includes
<b>IssueUpdateBatcher-subject.vm</b>	Email subject for batched issue updates.	-
<b>IssueUpdateBatched-header.vm</b>	Email header. This includes: <ul style="list-style-type: none"><li>• Text summarizing the updates (There is 1 comment)</li><li>• Issue details: Project, name, key, summary, actions</li></ul>	-
<b>IssueUpdateBatcher-content.vm</b>	Main part of the email where issue updates are shown. The updates themselves, however, are retrieved by the next template (below) which is included in this one.	<ul style="list-style-type: none"><li>• Templates: IssueUpdateBatcher-history.vm</li><li>• Templates: IssueUpdateBatcher-header.vm</li><li>• Includes: hr-bottom.vm</li><li>• Includes: footer.vm</li></ul>

<b>IssueUpdateBatcher-history.vm</b>	Provides the summary of issue updates: comments, updates, work logs, mentions, and issue status (moved, assigned, archived, deleted, etc.).	<ul style="list-style-type: none"> <li>• Templates: shared/mention-text.vm</li> </ul>
--------------------------------------	---	---

## Includes

List of includes referenced in main templates for issue notifications.

Directory: `templates/email-batch/html`

Template	Description	Included in
<b>hr-bottom.vm</b>	Generates the divider (horizontal line).	<ul style="list-style-type: none"> <li>• Template: IssueUpdateBatcher-content.vm</li> </ul>
<b>template.vm</b>	Root template. Provides the base of the HTML document that is sent as notification email.	<ul style="list-style-type: none"> <li>• Templates: IssueUpdateBatcher.vm</li> </ul>
<b>footer.vm</b>	Footer of batched issue notifications.	<ul style="list-style-type: none"> <li>• Template: IssueUpdateBatcher-content.vm</li> </ul>
<b>../shared/mention-text</b>	Adds an extra line of text to batched issue notifications if you were mentioned in an issue.	<ul style="list-style-type: none"> <li>• Template: IssueUpdateBatcher-history.vm</li> </ul>

## Other notifications

These notifications control other events like sharing filters or user management.

## Share templates

Directory: `templates/email/html`

Template	Description	Includes
<b>shareissue.vm</b>	Someone is sharing an issue with you.	<ul style="list-style-type: none"> <li>• Includes: header.vm</li> <li>• Includes: footer.vm</li> <li>• Patterns: involvedUsers.vm</li> <li>• Patterns: comment-top.vm</li> <li>• Patterns: issue-title.vm</li> <li>• Templates: mobileSingleColumnStyle.vm</li> </ul>

<b>sharejqlsearch.vm</b>	Someone is sharing a JQL search with you.	<ul style="list-style-type: none"> <li>• Includes: header.vm</li> <li>• Includes: footer.vm</li> <li>• Patterns: involvedUsers.vm</li> <li>• Patterns: comment-top.vm</li> <li>• Patterns: page-title.vm</li> <li>• Templates: mobileSingleColumnStyle.vm</li> </ul>
<b>sharesavedsearch.vm</b>	Someone is sharing a saved search (filter) with you.	<ul style="list-style-type: none"> <li>• Includes: footer.vm</li> <li>• Patterns: involvedUsers.vm</li> <li>• Patterns: comment-top.vm</li> <li>• Patterns: page-title.vm</li> <li>• Templates: mobileSingleColumnStyle.vm</li> </ul>

### Contact admin templates

Directory: templates/email/html

Template	Description	Includes
<b>contactadministrator.vm</b>	Notification that Jira admin is getting after someone tries to contact them over the Contact Admin form.	<ul style="list-style-type: none"> <li>• Includes: header.vm</li> <li>• Includes: footer.vm</li> <li>• Patterns: page-title.vm</li> <li>• Patterns: text-paragraph.vm</li> </ul>
<b>emailfromadmin.vm</b>	Emails sent to users directly from the Jira admin.	<ul style="list-style-type: none"> <li>• Includes: footer.vm</li> <li>• Patterns: page-title.vm</li> <li>• Patterns: text-paragraph.vm</li> </ul>

### User management templates

Directory: templates/email/html

Template	Description	Includes
----------	-------------	----------

<b>usercreated.vm</b>	New user was created. Notification for Jira admins.	<ul style="list-style-type: none"> <li>• Includes: header.vm</li> <li>• Includes: footer.vm</li> <li>• Includes: userdetails.vm</li> <li>• Patterns: button-action.vm</li> <li>• Patterns: page-title.vm</li> <li>• Patterns: text-paragraph.vm</li> </ul>
<b>usercreated-nopassword.vm</b>	New user was created in an external directory, and their password cant be changed. Notification for Jira admins.	<ul style="list-style-type: none"> <li>• Includes: header.vm</li> <li>• Includes: footer.vm</li> <li>• Includes: userdetails.vm</li> <li>• Patterns: page-title.vm</li> <li>• Patterns: text-paragraph.vm</li> </ul>
<b>usersignup.vm</b>	Notification users receive after signing up to Jira.	<ul style="list-style-type: none"> <li>• Includes: header.vm</li> <li>• Includes: footer.vm</li> <li>• Includes: userdetails.vm</li> </ul>
<b>forgotusername.vm</b>	Notification users receive if they forget their usernames.	<ul style="list-style-type: none"> <li>• Includes: header.vm</li> <li>• Includes: footer.vm</li> </ul>
<b>forgotpassword.vm</b>	Notification users receive if they forget their password.	<ul style="list-style-type: none"> <li>• Includes: header.vm</li> <li>• Includes: footer.vm</li> <li>• Includes: userdetails.vm</li> </ul>

**Other notifications: Includes**

List of includes referenced in main templates.

Directory: templates/email/html/includes

Template	Description	Included in
<b>emailconstants.vm</b>	Fonts and colors of headers, headings, and links.	<ul style="list-style-type: none"> <li>• macros.vm</li> <li>• footer.vm</li> <li>• header.vm</li> </ul>
<b>footer.vm</b>	Email footer.	<p>Share:</p> <ul style="list-style-type: none"> <li>• shareissue.vm</li> <li>• sharejqlsearch.vm</li> <li>• sharesavedsearch.vm</li> </ul> <p>Contact admin:</p> <ul style="list-style-type: none"> <li>• contactadministrator.vm</li> <li>• emailfromadmin.vm</li> </ul> <p>User management:</p> <ul style="list-style-type: none"> <li>• usercreated.vm</li> <li>• usercreated-nopassword.vm</li> <li>• usersignup.vm</li> <li>• forgotusername.vm</li> <li>• forgotpassword.vm</li> </ul>
<b>header.vm</b>	Email header.	<p>Share:</p> <ul style="list-style-type: none"> <li>• shareissue.vm</li> <li>• sharejqlsearch.vm</li> </ul> <p>Contact admin:</p> <ul style="list-style-type: none"> <li>• contactadministrator.vm</li> </ul> <p>User management:</p> <ul style="list-style-type: none"> <li>• usercreated.vm</li> <li>• usercreated-nopassword.vm</li> <li>• usersignup.vm</li> <li>• forgotusername.vm</li> <li>• forgotpassword.vm</li> </ul>
<b>userdetails.vm</b>	Details of a user: username, email address, display name.	<p>User management:</p> <ul style="list-style-type: none"> <li>• usercreated.vm</li> <li>• usercreated-nopassword.vm</li> <li>• usersignup.vm</li> <li>• forgotpassword.vm</li> </ul>

**Other notifications: Patterns**

Its unlikely that youll need to edit any of the pattern templates. They contain reusable components that are common for the Jira UI, like action buttons, titles, or text paragraphs. If you wanted to check them out anyway, you can find them in `templates/email/html/patterns`.

# Templates: Separate issue notifications and other events

The archive that you downloaded from Jira contains email templates from all available configuration options. Its easy to get lost in the files and their dependencies, especially that not all of them apply to your configuration. Use this page to understand the templates and the notifications they control.

## This page is for separate issue notifications in Jira

This page is for separate issue notifications (batched issue updates are disabled). If youre using batched issue notifications, see [Batched issue notifications](#).

[Not sure which notifications youre using?](#)

## Definitions

Were referring to templates and includes around this page. Since all of them are Velocity templates, heres the best way to understand them:

- **Templates:** Main templates used for notifications. Usually, a template like that is a notification that youre getting. Youd modify them if youd like to customize the looks of a specific notification rather than an item that is shared with all notifications, like footer.
- **Includes:** Includes arent used on their own, but are always referenced in main templates. Youd modify them if youd like to customize the looks of an item shared with all notifications rather than the notification itself. For example, changing footers or headers.

## Issue notifications

These templates control notifications about issues.

Directory: `templates/email/html`

## Issue lifecycle

Template	Description	Includes
<b>issuecreated.vm</b>	Issue was created.	<ul style="list-style-type: none"><li>• Patterns: issue-title.vm</li><li>• Fields: issuetype.vm</li><li>• Fields: affectsversions.vm</li><li>• Fields: assignee.vm</li><li>• Fields: attachments.vm</li><li>• Fields: components.vm</li><li>• Fields: createddate.vm</li><li>• Fields: duedate.vm</li><li>• Fields: environment.vm</li><li>• Fields: fixversions.vm</li><li>• Fields: labels.vm</li><li>• Fields: priority.vm</li><li>• Fields: reporter.vm</li><li>• Fields: securitylevel.vm</li><li>• Fields: timetracking.vm</li><li>• Patterns: text-paragraph.vm</li><li>• Patterns: comment-action.vm</li><li>• Includes: footer.vm</li></ul>
<b>issuedeleted.vm</b>	Issue was deleted.	<ul style="list-style-type: none"><li>• Patterns: issue-title.vm</li><li>• Includes: footer.vm</li></ul>

<b>issueassigned.vm</b>	Issue was assigned to someone.	<ul style="list-style-type: none"> <li>• Patterns: comment-top.vm</li> <li>• Patterns: issue-title.vm</li> <li>• Fields: changelog.vm</li> <li>• Patterns: comment-action.vm</li> <li>• Includes: footer.vm</li> </ul>
<b>issuresolved.vm</b>	Issue was resolved.	<ul style="list-style-type: none"> <li>• Patterns: comment-top.vm</li> <li>• Patterns: issue-title.vm</li> <li>• Patterns: comment-action.vm</li> <li>• Includes: footer.vm</li> <li>• Fields: changelog.vm</li> </ul>
<b>issueclosed.vm</b>	Issue was closed.	<ul style="list-style-type: none"> <li>• Patterns: comment-top.vm</li> <li>• Patterns: issue-title.vm</li> <li>• Patterns: comment-action.vm</li> <li>• Fields: changelog.vm</li> <li>• Includes: footer.vm</li> </ul>
<b>issuereopened.vm</b>	Issue was reopened.	-
<b>issueupdated.vm</b>	Issue was updated.	<ul style="list-style-type: none"> <li>• Patterns: issue-title.vm</li> <li>• Patterns: comment-top.vm</li> <li>• Patterns: comment-action.vm</li> <li>• Includes: footer.vm</li> <li>• Includes: changelog-issue-description.vm</li> <li>• Fields: changelog.vm</li> </ul>
<b>issuemoved.vm</b>	Issue was moved to a different project, issue type, etc.	<ul style="list-style-type: none"> <li>• Patterns: text-top.vm</li> <li>• Patterns: issue-title.vm</li> <li>• Patterns: comment-action.vm</li> <li>• Includes: footer.vm</li> </ul>
<b>issuegenericevent.vm</b>	Issue generic event	<ul style="list-style-type: none"> <li>• Patterns: comment-top.vm</li> <li>• Patterns: issue-title.vm</li> <li>• Patterns: comment-action.vm</li> <li>• Includes: changelog-issue-description.vm</li> <li>• Includes: footer.vm</li> <li>• Fields: changelog.vm</li> </ul>

### Issue collaboration

Template	Description	Includes
----------	-------------	----------

<b>issuecommented.vm</b>	Issue was commented on.	<ul style="list-style-type: none"> <li>• Patterns: comment-title.vm</li> <li>• Patterns: comment-action.vm</li> <li>• Includes: set-issue-details.context.vm</li> <li>• Includes: footer.vm</li> </ul>
<b>issuecommentedited.vm</b>	Issue comment was edited.	<ul style="list-style-type: none"> <li>• Patterns: comment-title.vm</li> <li>• Patterns: comment-top.vm</li> <li>• Patterns: comment-action.vm</li> <li>• Includes: set-issue-details-context.vm</li> <li>• Includes: footer.vm</li> </ul>
<b>issuementioned.vm</b>	Someone mentioned you in an issue.	<ul style="list-style-type: none"> <li>• Patterns: comment-title.vm</li> <li>• Patterns: comment-top.vm</li> <li>• Patterns: comment-action.vm</li> <li>• Includes: set-issue-details-context.vm</li> <li>• Includes: footer.vm</li> </ul>
<b>issuenotify.vm</b>		<ul style="list-style-type: none"> <li>• Patterns: comment-top.vm</li> <li>• Patterns: issue-title.vm</li> <li>• Patterns: comment-action.vm</li> <li>• Includes: footer.vm</li> <li>• Fields: changelog.vm</li> </ul>

### Issue archiving

Template	Description	Includes
<b>issuearchived.vm</b>	Issue was archived.	<ul style="list-style-type: none"> <li>• Patterns: issue-title.vm</li> <li>• Includes: footer.vm</li> </ul>
<b>issuerestored.vm</b>	Issue was restored from the archive.	<ul style="list-style-type: none"> <li>• Patterns: issue-title.vm</li> <li>• Includes: footer.vm</li> </ul>

### Issue work logging

Template	Description	Includes
<b>issueworklogdeleted.vm</b>	Issue work log was deleted.	-
<b>issueworklogged.vm</b>	Work was logged in an issue.	-
<b>issueworklogupdated.vm</b>	Issue work log was updated.	-

<b>issueworkstarted.vm</b>	Work started on issue.	<ul style="list-style-type: none"> <li>• Patterns: comment-top.vm</li> <li>• Patterns: comment-action.vm</li> <li>• Fields: changelog.vm</li> <li>• Includes: set-issue-details-context.vm</li> <li>• Includes: footer.vm</li> </ul>
<b>issueworkstopped.vm</b>	Work stopped.	<ul style="list-style-type: none"> <li>• Patterns: comment-top.vm</li> <li>• Patterns: comment-action.vm</li> <li>• Includes: set-issue-details-context.vm</li> <li>• Includes: footer.vm</li> <li>• Fields: changelog.vm</li> </ul>

## Other notifications

These notifications control other events like sharing filters or user management.

## Share templates

Directory: `templates/email/html`

Template	Description	Includes
<b>shareissue.vm</b>	Someone is sharing an issue with you.	<ul style="list-style-type: none"> <li>• Includes: header.vm</li> <li>• Includes: footer.vm</li> <li>• Patterns: involvedUsers.vm</li> <li>• Patterns: comment-top.vm</li> <li>• Patterns: issue-title.vm</li> <li>• Templates: mobileSingleColumnStyle.vm</li> </ul>
<b>sharejqlsearch.vm</b>	Someone is sharing a JQL search with you.	<ul style="list-style-type: none"> <li>• Includes: header.vm</li> <li>• Includes: footer.vm</li> <li>• Patterns: involvedUsers.vm</li> <li>• Patterns: comment-top.vm</li> <li>• Patterns: page-title.vm</li> <li>• Templates: mobileSingleColumnStyle.vm</li> </ul>
<b>sharesavedsearch.vm</b>	Someone is sharing a saved search (filter) with you.	<ul style="list-style-type: none"> <li>• Includes: footer.vm</li> <li>• Patterns: involvedUsers.vm</li> <li>• Patterns: comment-top.vm</li> <li>• Patterns: page-title.vm</li> <li>• Templates: mobileSingleColumnStyle.vm</li> </ul>

## Contact admin templates

Directory: `templates/email/html`

Template	Description	Includes	
----------	-------------	----------	--

<b>contactadministrator.vm</b>	contactadministrator.vm	Notification that Jira admin is getting after someone tries to contact them over the Contact Admin form.	<ul style="list-style-type: none"> <li>• Includes: header.vm</li> <li>• Includes: footer.vm</li> <li>• Patterns: page-title.vm</li> <li>• Patterns: text-paragraph.vm</li> </ul>
<b>emailfromadmin.vm</b>	emailfromadmin.vm	Emails sent to users directly from the Jira admin.	<ul style="list-style-type: none"> <li>• Includes: footer.vm</li> <li>• Patterns: page-title.vm</li> <li>• Patterns: text-paragraph.vm</li> </ul>

### User management templates

Directory: `templates/email/html`

Template	Description	Includes
<b>usercreated.vm</b>	New user was created. Notification for Jira admins.	<ul style="list-style-type: none"> <li>• Includes: header.vm</li> <li>• Includes: footer.vm</li> <li>• Includes: userdetails.vm</li> <li>• Patterns: button-action.vm</li> <li>• Patterns: page-title.vm</li> <li>• Patterns: text-paragraph.vm</li> </ul>

<b>usercreated-nopassword.vm</b>	New user was created in an external directory, and their password cant be changed. Notification for Jira admins.	<ul style="list-style-type: none"> <li>• Includes: header.vm</li> <li>• Includes: footer.vm</li> <li>• Includes: userdetails.vm</li> <li>• Patterns: page-title.vm</li> <li>• Patterns: text-paragraph.vm</li> </ul>
<b>usersignup.vm</b>	Notification users receive after signing up to Jira.	<ul style="list-style-type: none"> <li>• Includes: header.vm</li> <li>• Includes: footer.vm</li> <li>• Includes: userdetails.vm</li> </ul>
<b>forgotusername.vm</b>	Notification users receive if they forget their usernames.	<ul style="list-style-type: none"> <li>• Includes: header.vm</li> <li>• Includes: footer.vm</li> </ul>
<b>forgotpassword.vm</b>	Notification users receive if they forget their password.	<ul style="list-style-type: none"> <li>• Includes: header.vm</li> <li>• Includes: footer.vm</li> <li>• Includes: userdetails.vm</li> </ul>

## Includes

### Includes

List of includes referenced in main templates.

Directory: `templates/email/html/includes`

Template	Description	Included in
<b>emailconstants.vm</b>	Fonts and colors of headers, headings, and links.	<ul style="list-style-type: none"> <li>• macros.vm</li> <li>• footer.vm</li> <li>• header.vm</li> </ul>

<b>footer.vm</b>	Email footer.	<p>Share:</p> <ul style="list-style-type: none"> <li>• shareissue.vm</li> <li>• sharejqlsearch.vm</li> <li>• sharesavedsearch.vm</li> </ul> <p>Contact admin:</p> <ul style="list-style-type: none"> <li>• contactadministrator.vm</li> <li>• emailfromadmin.vm</li> </ul> <p>User management:</p> <ul style="list-style-type: none"> <li>• usercreated.vm</li> <li>• usercreated-nopassword.vm</li> <li>• usersignup.vm</li> <li>• forgotusername.vm</li> <li>• forgotpassword.vm</li> </ul>
<b>header.vm</b>	Email header.	<p>Share:</p> <ul style="list-style-type: none"> <li>• shareissue.vm</li> <li>• sharejqlsearch.vm</li> </ul> <p>Contact admin:</p> <ul style="list-style-type: none"> <li>• contactadministrator.vm</li> </ul> <p>User management:</p> <ul style="list-style-type: none"> <li>• usercreated.vm</li> <li>• usercreated-nopassword.vm</li> <li>• usersignup.vm</li> <li>• forgotusername.vm</li> <li>• forgotpassword.vm</li> </ul>
<b>userdetails.vm</b>	Details of a user: username, email address, display name.	<p>User management:</p> <ul style="list-style-type: none"> <li>• usercreated.vm</li> <li>• usercreated-nopassword.vm</li> <li>• usersignup.vm</li> <li>• forgotpassword.vm</li> </ul>
<b>changelog-issue-description.vm</b>	Shows the old and updated issue description.	<ul style="list-style-type: none"> <li>• issueupdated.vm</li> <li>• issuegenericevent.vm</li> </ul>

### Issue notifications: Fields

Issue fields referenced in main templates to display data from specific fields. Applies only to issue notification templates.

Directory: `templates/email/html/fields`

**Other notifications: Patterns**

Its unlikely that youll need to edit any of the pattern templates. They contain reusable components that are common for the Jira UI, like action buttons, titles, or text paragraphs. If you wanted to check them out anyway, you can find them in `templates/email/html/patterns`.

Directory: `templates/email/html/patterns`

# Examples: Customizing email content

Here are some examples of how to modify your email templates.

## Adding customized onboarding instructions

**Template:** usersignup.vm

The usersignup.vm template controls notifications sent to new users. You can add your onboarding instructions, or external links, somewhere before the footer, as shown in this example:

```
#rowWrapperNormalBegin()  
  
<Your onboarding instructions>  
  
#rowWrapperNormalEnd()  
  
#parse("templates/email/html/includes/footer.vm")
```

## Adding company logo

**Template:** any, most commonly footer.vm or header.vm

1. Add the image file to the ZIP archive you downloaded from Jira. You should create a new folder for it at the top of the hierarchy, next to email and email-batch.
2. Refer to the image file in your templates. Jira will send the image together with notifications.

```

```

## Showing issues URL instead of content

**Template:** Issue notifications templates (batched and separate)

Use the following snippet to generate a URL to your issue. In this URL, the issue summary will be displayed as URL text. If you don't want to reveal the summary, you can change it to something else.

```
<a href='${baseurl}/browse/${issue.getKey()}'>${textutils.htmlEncode($issue.getSummary())}</a>
```

Once you have the URL, use it in one of the templates for issue notifications. For example, if you wanted to modify issuecreated.vm, you'd have to replace its content with the following snippet:

```
#disable_html_escaping()  
  
#defaultMailHeader("email.event.activity.created.issue", $issue.reporter)  
  
#set($link = "<a href='${baseurl}/browse/${issue.getKey()}'>${textutils.htmlEncode($issue.getSummary())</a>")  
  
#rowWrapperNormal($link)  
  
#parse("templates/email/html/includes/footer.vm")
```

## Displaying conditional content based on security level

**Template:** Issue notifications templates (batched and separate)

1. Add the following Velocity snippet to define the name of your security level:

```
#if ($issue.securityLevel)
  #(set($issueSecurityLevel = $issue.securityLevel.getString(name))
#end
```

2. Display your content if the issue matches the defined security level:

```
#if($issueSecurityLevel == "My Security Level")
  The content you'd like to display for this security level.
#end
```

# Customizing email content in Jira Server

Jira generates emails in reaction to events using a templating engine - Apache's [Velocity](#). This is a relatively easy to use templating language that can pull apart java objects in useful ways. Whether its adding some info to the header or footer, or shuffling around the contents of your emails, you can make a number of customizations and tailor the emails to your needs.



Customizations to Velocity templates or other Jira files are not included in the scope of [Atlassian Support](#).

## Jump to...

- [Before you begin](#)
- [Retrieving templates for batched issue notifications](#)
- [Retrieving templates for separate issue notifications \(and other events\)](#)

## Before you begin

Email templates in Jira Server are stored in two locations:

- **Jira installation directory:** Batched issue notifications
- **Jira inform - batchers app:** Separate issue notifications and other events

Which ones you need to retrieve depends on what kind of notifications are you looking to customize. If youre not sure which notifications youre using, see [Different types of email notifications in Jira](#).



This page is for Jira Server. If youre using Jira Data Center, you will retrieve the templates in a different way than described here. See [Customizing email content](#).

## Retrieving templates for batched issue notifications

Retrieve these templates to customize batched issue notifications.

### Good to know

- You don't have to restart Jira after modifying these templates.
- The app that contains the templates will get overwritten with every Jira upgrade, and you'll need to reapply your changes. Its good to keep the copy of the templates somewhere, so you can reapply the changes more quickly (don't just copy the whole app as its version may change).

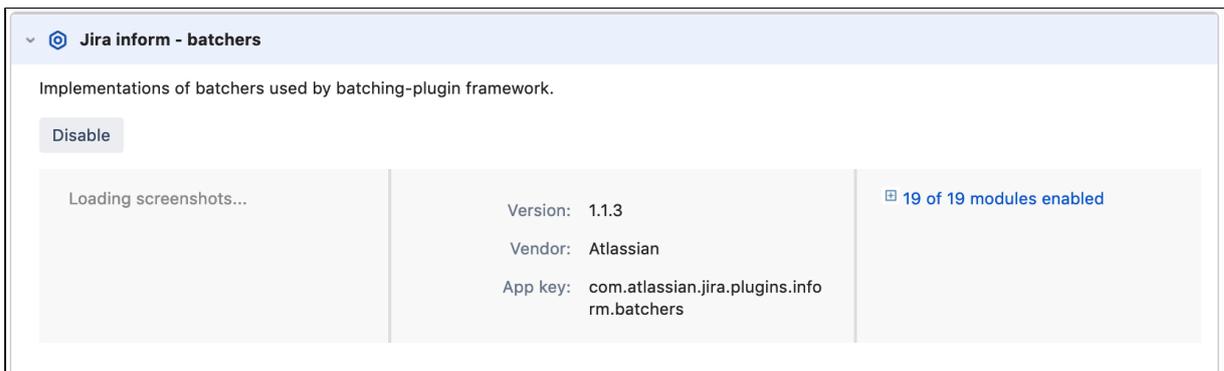
### Step 1: Retrieve the Velocity templates

The Velocity templates used for batched issue notifications are in the **Jira inform - batchers** app. Youll need to extract them.

#### 1. Find out the app version

Your Jira installation directory might have more versions of this app. Check your current version to know which file to edit later.

1. Go to **Administration > Manage apps**, and open the **Manage apps** page.
2. From the drop-down, select **All apps**, and search for **Jira inform - batchers**.
3. Expand the app, and check your version (in the example below, its 1.1.3).



## 2. Copy the app from the Jira installation directory

Copy the app to a separate directory. You should never edit the .jar files inside the Jira installation directory. Its also good to keep the original JAR file in case you needed to revert the changes.

1. Go to <jira-installation-directory>/atlassian-jira/WEB-INF/atlassian-bundled-plugins/.
2. Find the *batchers-<version>.jar* file.
3. Copy the file to a separate directory.

## 3. Extract the template files from the apps JAR file

The easiest way to extract the templates is to use the following command. Note that youll need JDK installed for it to work.

```
jar xf batchers-1.1.3.jar templates/email
```

## Step 2: Edit the Velocity templates

Once youve extracted the Velocity templates, you can edit them directly. For more info about the templates and what you can do with them, see:

- [Templates: Batched issue notifications](#)

## Step 3: Upload the updated templates to Jira

### Testing your changes

We recommend that you test your changes in a staging environment before applying them in production. If you break the Velocity syntax, emails wont be sent at all.

1. Insert the templates back into the JAR file.

```
jar uf batchers-1.1.3.jar templates/email
```

2. Upload the app to Jira.
  - a. Go to **Administration > Manage apps > Manage apps**.
  - b. Click **Upload app**, and upload the JAR file. The changes will be visible once the app is reinstalled. You dont have to restart your Jira instance.

## Retrieving templates for separate issue notifications (and other events)

Retrieve these templates to customize separate issue notifications, or any other notifications, like User created, Forgot Password, Contact admin, etc.

## Good to know

- Any changes to Velocity templates (as well as JSPs) will be overwritten following a Jira upgrade. In such a case, you will need to manually copy your modified files to the new Jira version. If the Velocity templates and/or JSPs have changed in the newer version, you will have to manually port your customizations into them (as opposed to copying these files directly over from your old Jira installation to your upgraded one).

## Step 1: Retrieve the Velocity templates

1. Go to the following location in your Jira installation directory:

```
/atlassian-jira/WEB-INF/classes/templates/email/
```

If you're using Jira sources files, go to `jira/src/etc/java/templates/email/` instead.

2. Under this directory, there are three subdirectories:

- `html`- contains the templates used to create emails in html
- `text`- contains the templates used to create plain text mail outs
- `subject`- contains the templates used to generate the subject of the emails

## Creating new email templates

1. Create your new `mytemplate.vm` files in the `html`, `text` and `subject` directories, based on the existing files in those directories
2. Add the templates to `atlassian-jira/WEB-INF/classes/email-template-id-mappings.xml` to make them valid choices for when you are adding a new event.

## Step 2: Edit the Velocity templates

Once you've extracted the Velocity templates, you can edit them directly. For more info about the templates and what you can do with them, see:

- [Templates: Separate issue notifications and other events](#)

## Step 3: Restart Jira

After modifying these templates, you'll need to restart your Jira instance to apply the changes.

### Optional: Deploying Velocity templates without restarting Jira

In a *development* instance, you can play with picking up velocity file changes without a restart. From `<jira-install>/atlassian-jira/WEB-INF/classes/velocity.properties`:

1. Change `class.resource.loader.cache` from `true` to `false`
2. Remove the comment sign (`#`) from `#velocimacro.library.autoreload=true`

Making this change in production will eventually lead to Jira not serving pages along with the 'ran out of parsers' error in the log file.

# Adding custom fields to emails

This tutorial will help you add a custom field to batched issue notifications. If you're looking to add custom fields to separate issue notifications, see [Adding custom fields to separate issue notifications](#) instead.

 This tutorial is for advanced users, and is out of scope of Atlassian support. You will be adding custom fields by editing the code in the Velocity templates, which email notifications are based on.

## Overview

Notifications inform you about changes in the issue's fields (both built-in and custom fields). If a field's value did not change, it won't be included in the notification, because there's nothing to be notified about. After all, notifications are all about showing a change.

Here you can learn how to include a custom field, and its current value, in every notification for an issue, even if this field wasn't updated. You can use it to describe the issues you're being notified about more precisely. For example, you can include a custom field that specifies the issue's security level, and then properly categorize or even hide the notifications for this issue.

## Before you begin

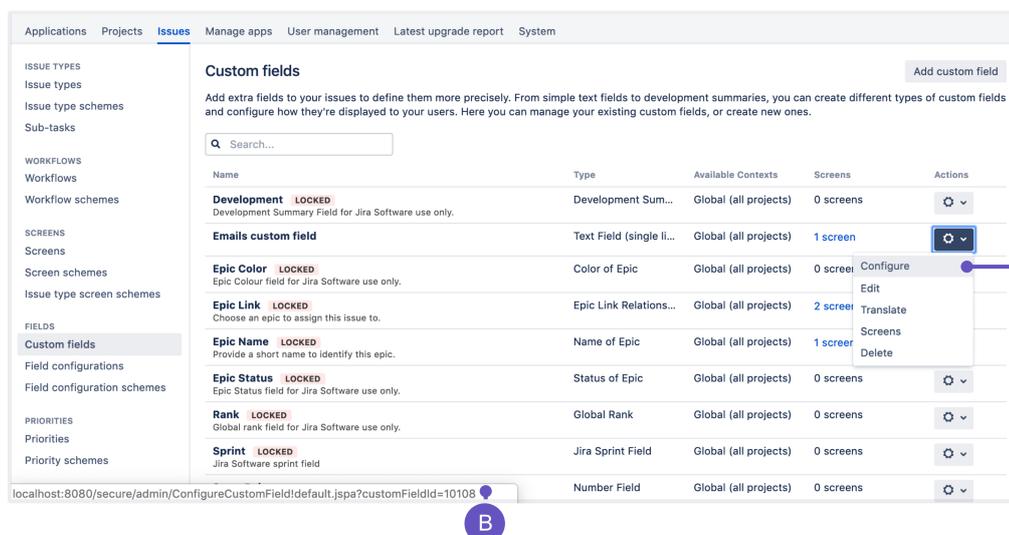
- [Add your custom field](#) to an issue.
- **Limitation:** Adding extra issue fields to your emails is not supported for batched notifications. You can only add custom fields. If you want to have issue fields displayed, you'll need to switch to separate notifications, where these fields are supported.

## Step 1: Find the ID of your custom field

You will add custom fields to your email templates by using their IDs. You can find the ID by examining the URLs of custom fields or by querying the database.

## Examining the URL of a custom field

1. Go to **Administration > Issues**, and open the **Custom fields** page.
2. Find your custom field, and click **Actions** (  ) to see more options.



Name	Type	Available Contexts	Screens	Actions
<b>Development</b> <small>LOCKED</small> Development Summary Field for Jira Software use only.	Development Sum...	Global (all projects)	0 screens	
<b>Emails custom field</b>	Text Field (single li...	Global (all projects)	1 screen	
<b>Epic Color</b> <small>LOCKED</small> Epic Colour field for Jira Software use only.	Color of Epic	Global (all projects)	0 screens	
<b>Epic Link</b> <small>LOCKED</small> Choose an epic to assign this issue to.	Epic Link Relations...	Global (all projects)	2 screens	
<b>Epic Name</b> <small>LOCKED</small> Provide a short name to identify this epic.	Name of Epic	Global (all projects)	1 screen	
<b>Epic Status</b> <small>LOCKED</small> Epic Status field for Jira Software use only.	Status of Epic	Global (all projects)	0 screens	
<b>Rank</b> <small>LOCKED</small> Global rank field for Jira Software use only.	Global Rank	Global (all projects)	0 screens	
<b>Sprint</b> <small>LOCKED</small> Jira Software sprint field	Jira Sprint Field	Global (all projects)	0 screens	
	Number Field	Global (all projects)	0 screens	

**A.** Hover your mouse over the **Configure** item in the drop-down menu. The URL will display in the footer of your browser.

**B.** Here's the custom fields ID. In this example, it's 10108.

## Querying the database

Run the following query on your database:

```
SELECT * FROM customfield WHERE cfname LIKE '%mycustomfield%';
```

Where `mycustomfield` is the name of your custom field, for example `assignee`.

## Step 2: Add the custom field to the Velocity context

Before you can add custom fields to the email templates, you need to define them in the Velocity context. These steps might require some knowledge about REST API. If you're having problems, see [Jira REST API](#).

### 1. Retrieve currently defined custom fields

You can use this command:

```
curl -D- \
-u username:password \
http://localhost:8080/rest/inform-batchers/1.0/customfields
```

If you haven't added any custom fields yet, the list should be empty, like in the following example:

```
{
  "customFieldIds": []
}
```

### 2. Add your custom field

To add your custom field to the Velocity context, you can use the following command. Replace `<ID>` with the ID of your custom field.

```
curl \
-D- \
-u username:password \
-X POST \
-H "Content-Type: application/json" \
http://localhost:8080/rest/inform-batchers/1.0/customfields?id=customfield_<ID>
```

The result of this command should look similar to this:

```
{
  "customFieldIds": ["customfield_10108"]
}
```

### Removing a custom field

You can remove any of the custom fields from the Velocity context by using this command:

```
curl \
-D- \
-u username:password \
-X DELETE \
-H "Content-Type: application/json" \
http://localhost:8080/rest/inform-batchers/1.0/customfields?id=customfield_<ID>
```

## Step 3: Retrieve the Velocity templates

Retrieve your email templates so you can make the changes.

- For Jira Data Center, see [Customizing email content](#).
- For Jira Server, see [Customizing email content in Jira Server](#).

## Step 4: Edit the Velocity templates

Once you've extracted the Velocity templates, you can edit them directly to add code snippets that will display your custom fields.

### 1. Choose a template to update

Batched email notifications are using several templates. If you don't know which one to edit, see [Templates: Batched issue notifications](#).

### 2. Edit the template

Jira supports html and text email formats. You should choose instructions according to format set in your Jira.

1. Find the Velocity template of the email type you wish to modify.
2. Add the following snippet where you want it to appear in the file:

```
#if($customFields.get('customfield_<ID>').getValue())
${customFields.get('customfield_<ID>').getName(): ${customFields.get('customfield_<ID>').
getValue()}
#end
```

1. Find the Velocity template of the email type you wish to modify.
2. Add the following snippet where you want it to appear in the file:

```
#if(${customFields.get('customfield_<ID>').getValue()})
<tr>
  <td>${escape.apply($customFields.get('customfield_<ID>').getName()):</td>
  <td>
    ${escape.apply($customFields.get('customfield_<ID>').getValue())
  </td>
</tr>
#end
```

### Some tips for editing this code snippet:

- This is where you enter the ID of your custom field.

```
${customFields.get('customfield_<ID>').getValue() }
```

Optionally, you can replace this line with one of the two below:

- Use this code to retrieve the most up-to-date value of your custom field. This is useful when a custom field gets deleted, in which case the value is returned as null. In the original line, you'd get the last value before the deletion.

```
${customFieldsCurrent.get('customfield_<ID>').getValue() }
```

- Use this code to retrieve the name of your custom field.

```
${customFields.get('customfield_<ID>').getName() }
```

## Step 5: Upload the updated templates to Jira

 **Testing your changes**

We recommend that you test your changes in a staging environment before applying them in production. If you break the Velocity syntax, emails wont be sent at all.

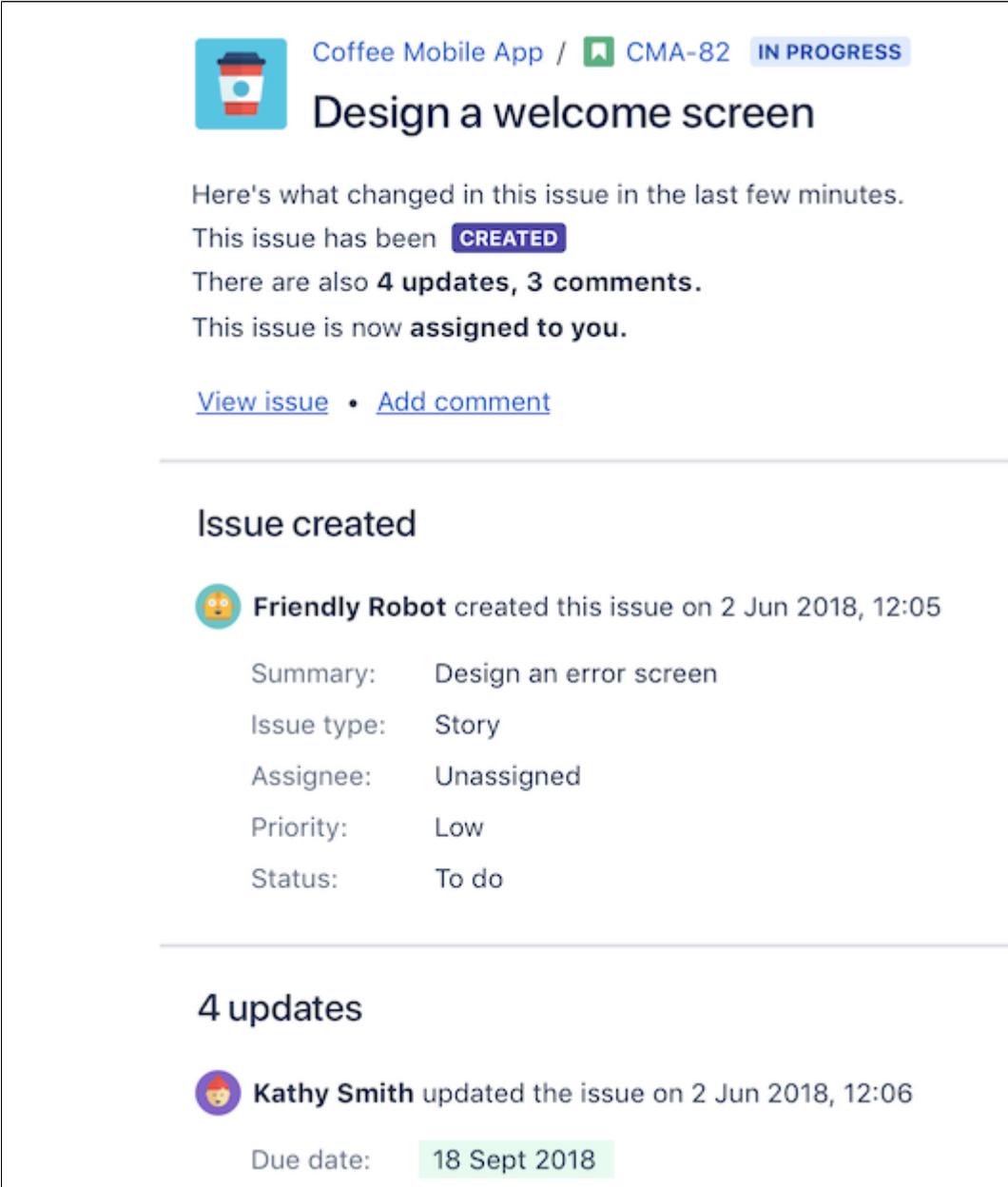
- For Jira Data Center, see [Customizing email content](#).
- For Jira Server, see [Customizing email content in Jira Server](#).

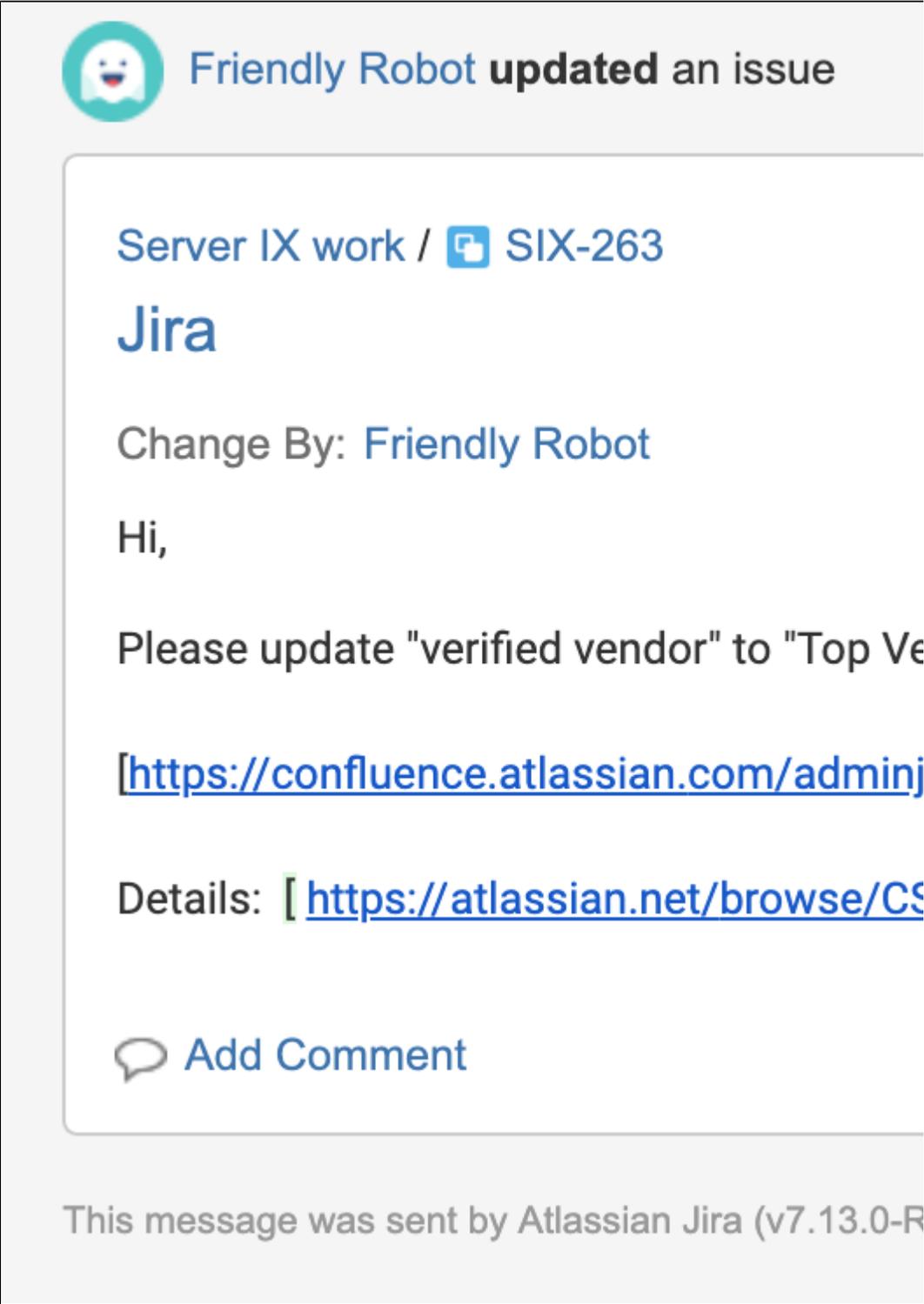
# Different types of email notifications in Jira

When reading about email notifications, you might have seen references to *batched*, *separate*, and even *them* otifications. This page is here to tell you the difference.

## Notifications about issues

The difference really only applies to notifications about issues in Jira. It came to life when Jira introduced *batched issue notification* that aimed to reduce the amount of emails coming from Jira by grouping issue updates that occurred close together into a single summary email.

Type	Jira version	Description
Batched issue notifications	8.1 + Default from Jira 8.4	<p>Issue updates that occur close together are grouped and sent in a single email.</p> <div></div>

Separate issue notifications	Any Default up to Jira 8.4	<p>Every issue update is sent in a separate email.</p> 
------------------------------	-------------------------------	--

### Still not sure?

You can also check which notifications your Jira instance is using by going to **Administration > System > Batched email notifications**. If the feature is on, you're using *batched issue notifications*.

### Other notifications

Notifications about other events don't have different types as they always come in separate emails. This includes notifications like User created, User signup, Forgot password, Contact admin, Email from admin, etc. To put it shortly, it's every notification that is not related to updates in your issues.

# Creating issues and comments from email

Admins can configure Jira to receive and process emails. Jira can receive emails from licensed users to create issues or add comments and attachments to existing issues automatically.



If you're looking for a help desk solution, it may be more practical to use Jira Service Management, rather than setting up Jira Core or Jira Software for this purpose.

Jira Service Management uses a built-in processor to receive and process issue requests from emails. Issues created in Jira Service Management don't require the sender to have a license to create, view, comment, add attachments, or transition issues. [Read more about receiving email requests with Jira Service Management](#).

[Learn how to download and set up Jira Service Management for your instance.](#)

## On this page:

- [Configuring issue or comment creation from email](#)
- [Mail handlers](#)
- [Issue/comment creation](#)
- [Handy tips with mail handlers](#)
- [Best practices \(pre-processing Jira email messages\)](#)
- [Troubleshooting](#)

## Configuring issue or comment creation from email

Issues and comments in Jira can be generated either from:

- email messages sent to an account on a POP or IMAP mail server, or
- messages written to the file system generated by an external mail service.



Note that for all of the following procedures, you must be logged in as a user with the **Jira Administrators** [global permission](#).

### Step one: Configure OAuth 2.0 integration

This is mandatory if you have been or are planning on using Gmail or Microsoft Exchange Online. Both Google and Microsoft will disable Basic Authentication in the upcoming future, and once this happens, you will not be able to create issues and comments from email and your connection to the Gmail and/or Microsoft Exchange Online server will no longer be operational.

You do not need to update the settings in your custom email servers or other service providers if they use IMAP or POP3. They will continue to work.

For Google and Microsoft, you need to configure the OAuth 2.0 settings first. To do that you will require specific info such as a client ID from your service provider. You can generate this data on the service providers side. Then, you need to copy the data to the OAuth plugin in your application to generate a redirect URL. You need to provide the redirect URL that your application generated at the service providers site. Once you save your configuration, you can proceed to configuring your mail server to use OAuth 2.0 as the authentication method.

For more information, see [Integrating with OAuth 2.0](#)

### Step two: Configure a mail server/service

#### POP or IMAP email messages

To set up issue and comment creation from email, you will need to create a mail account for a POP or IMAP mail server that Jira can access typically, one mail account for each Jira project. For example, for the 'ABC' project, you might establish an account [abc-issues@example.com](mailto:abc-issues@example.com)

Jira will periodically scan for new email messages received by your mail account (via a service) and appropriately create issues or comments for any emails it finds (via a mail handler).

Jira's [mail handlers](#) can also *optionally* create new user accounts for senders not previously seen. See the [Create a new issue or add a comment to an existing issue](#) section for more details.

⚠ Note that this is not possible if you are using [External User Management](#).

Once you have created a mail account on a POP or IMAP mail server, [configure Jira to receive email from that mail server account](#).

✔ **Tip:** You can configure Jira's mail servers so that recipients of email notifications can simply reply to these messages and have the body of their replies added as comments to the relevant issue. To do this, simply set the **From address** in [Jira's SMTP mail server](#) to match that of the POP or IMAP mail server's account being monitored. In most cases, this means having Jira's SMTP and POP or IMAP mail servers use the same mail account. See [below](#) for details on how to configure Jira to handle these emailed replies.

### File system messages

To set up issue and comment creation from messages written to the file system by an external mail service, your external mail service must be able to write these messages within the `import/mail` subdirectory of the [Jira home directory](#).

External mail services are very much like the POP or IMAP services above, except that instead of email messages being read from a mail account, they are read from a directory on the disk. External mail services are useful because they overcome the potential security risks associated with anonymous mail accounts. Instead you can simply configure your external mail service to dump incoming email messages within the Jira Home Directory's `import/mail` subdirectory, which is scanned periodically.

Please also be aware that Jira expects only one message per file, so your external mail service should be configured to generate such output.

📘 Note how Jira handles messages on a mail server/service:

- For mail accounts, Jira scans email messages received by your mail account's 'Inbox' folder. However, for IMAP mail servers, you can specify a different folder within your mail account.
- If Jira successfully processes a message, Jira deletes the message from your mail account (on a POP) or file system (i.e. for file system messages). On an IMAP mail server processed messages are not deleted but marked as read.
- If Jira does not successfully process a message, the message will remain either in your mail account or on the file system.

### Step two: Configure a mail handler

Once you have configured Jira to receive messages from a mail server/service, you configure Jira to handle these messages through a 'mail handler'.

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. Select **Mail > Incoming Mail** to open the Incoming Mail page.

- Click the **Add incoming mail handler** button (or the **Edit** link next to an existing mail handler) in the **Mail Handlers** section to open the **Mail Handler** dialog box.

- Specify a **Name** that describes what your mail handler will do for example, 'Create issues or comments from Example Company's IMAP mail server'.
- Select the mail **Server** that you configured in step one (above). This is either a POP or IMAP mail server or the **Local Files** option for an external mail service that writes messages to the file system.
- Specify the **Delay** (in minutes) between the mail handler's running time. This effectively defines the frequency with which Jira scans the **Server** that you specified in the previous step.
- Choose the type of mail **Handler** from dropdown list. For more information, refer to the [Mail Handlers](#) section below.
- If you chose either an IMAP mail server or the **Local Files** option in the **Server** field, then a **Folder Name** field appears below the **Handler** dropdown list:
  - For an IMAP mail server, if you want mail handler to scan for new messages from a folder other than the 'Inbox' in your mail account, specify the name of that folder here.
  - For the **Local Files** option, if your file messages are being written to a subdirectory within the `import/mail` subdirectory of the Jira home directory, specify the subdirectory structure (within `import/mail`) here.
- Click **Next** to continue with specifying the remaining options specific to mail **Handler** you selected above. For more information, see the [Mail Handlers](#) section below.
- (Optional) Click the **Test** button to test your mail handler. If you are using Local Files as the server, copy a saved email that contains a "Subject: " line to the configured directory. Jira will remove this file after it is parsed, or log a message about why an issue could not be created. You may have to specify the project, issuetype and reporterusername properties as a minimum configuration. A sample email file might look like this:  
 To: [jira@example.com](mailto:jira@example.com)  
 From: [some-jira-user@example.com](mailto:some-jira-user@example.com)  
 Subject: (TEST-123) issue summary title here  
 Body of the email goes here
- Click the **Add / Save** button to save your mail handler.

**Note the relationship between Jira mail handlers and services:**

- A Jira mail handler is part of a [Jira service](#). Hence, when you create a mail handler, its service will appear as an entry on the Services page.
- Be aware that editing mail handlers can only be performed through the Mail Handlers page (described above).
- On the Mail Handlers page, clicking the Delete link associated with a mail handler removes that handler. Since a mail handler is part of a service, then if you delete a mail handler's service on the Services page, its associated handler will also be removed from the Mail Handlers page.

## Mail handlers

Jira provides the following default mail handlers:

- [Create a new issue or add a comment to an existing issue](#)
- [Add a comment from the non quoted email body](#)
- [Add a comment with the entire email body](#)
- [Create a new issue from each email message](#)
- [Add a comment before a specified marker or separator in the email body](#)

For more information about how these mail handlers create issues and comments in Jira, refer to [Issue /comment creation](#) (below).

Also refer to the [Handy tips with mail handlers](#) (below) for tips on tweaking mail handlers to allow Jira to handle the following types of email messages:

- Email sent from people without a Jira user account.

### Create a new issue or add a comment to an existing issue

 This email handler doesn't use regex. We advise using the **Add a comment before a specified marker or separator in the email body** handler instead. For more information, see [Set up Jira notifications](#).

This message handler creates a new issue, or adds a comment to an existing issue. If the subject contains an issue key, the message is added as a comment to that issue. If no issue key is found, a new issue is created in the default project.

#### To configure a 'Create a new issue or add a comment to an existing issue' mail handler:

1. If you have not already done so, begin configuring your mail handler ([above](#)).
2. On the **Create a new issue or add a comment to an existing issue** dialog box, complete the following fields/options:

<b>Project</b>	<p>Specify the project key of the default project to which new issues are created by this handler for example, JIRA.</p> <p> <b>Note:</b></p> <ul style="list-style-type: none"> <li>• This field is only relevant for issue creation, not for issue commenting.</li> <li>• If an email message contains an issue key in its subject line and that issue key exists in your Jira installation, the handler will add the email message content as a comment on the issue, regardless of which project the issue is in.</li> </ul>
<b>Issue Type</b>	Choose the default issue type for new issues.
<b>Strip Quotes</b>	Select this checkbox to remove quoted text from from an email message's body (e.g. from previous email replies) before the body's content is added to the Jira issue's comment.
<b>Catch Email Address</b>	<p>If specified, only email messages whose <b>To:</b>, <b>Cc:</b>, <b>Bcc:</b> lines contain the recipient specified in this field will be processed for example, <a href="mailto:issues@mycompany.com">issues@mycompany.com</a></p> <p> Make sure <b>not</b> to put the Catch email address in the Bcc header when sending email but rather place it in the <b>To:</b> and <b>Cc:</b> headers. This is because mail servers tend to strip the Bcc header. Additionally, the Mail handler will look for the Catch email address in this header <b>only</b> if the header is available in the email source.</p>

	<p>Upon specifying an address here, all email messages whose <b>To:</b>, <b>Cc:</b>, <b>Bcc:</b> lines contain addresses other than the <b>Catch Email Address</b> are ignored. This is useful if you have multiple aliases for the same mail account (e.g. <code>foo-support@example-co.com</code> and <code>bar-support@example-co.com</code> aliases for <code>support@example-co.com</code>) for multiple mail services (e.g. each one to create issues in separate Jira projects).</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p> <b>Note:</b> in practice, <b>this option is rarely useful</b> and should not be confused with the more common <b>Default Reporter</b>. You can only specify one catch email address and one issue type per mail handler.</p> <p>In addition, there is a known bug in Jira 7.0.0 and JIRA 7.0.1, which means that multiple email handlers that are used to create issues in different projects when an email is sent to multiple aliases will not process the email correctly. This has been fixed in JIRA 7.0.2. For more information, see</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> <p> <a href="#">JRASERVER-41834</a> - Duplicate issues creation fails - Creating multiple issues by one Email <span style="float: right;">CLOSED</span></p> </div> </div>
<p><b>Bulk</b></p>	<p>This option only affects 'bulk' email messages whose header has either its <b>Precedence:</b> field set to <b>bulk</b> or its <b>Auto-Submitted</b> field not set to <b>no</b>. Such messages would typically be sent by an automated service. When such an email message is received, the following action will be performed, based on the option you choose:</p> <ol style="list-style-type: none"> <li>a. Ignore the email and do nothing.</li> <li>b. Forward the email (i.e. to the address set in the <b>Forward Email</b> text field).</li> <li>c. Delete the email permanently.</li> </ol> <p>It is generally a good idea to set <b>bulk=forward</b> and set a Forward Email address, to prevent mail loops between Jira and another automated service (eg. another Jira installation).</p>
<p><b>Forward Email</b></p>	<p>If specified, then if this mail service is unable to handle an email message it receives, an email message indicating this problem will be forwarded to the email address specified in this field.  <b>Note:</b> An <a href="#">SMTP mail server</a> must be configured for this option to function correctly.</p>
<p><b>Create Users</b></p>	<p>Select this checkbox if you want Jira to create new user accounts from any received email messages whose <b>From:</b> field contains an address that does not match one associated with an existing Jira user account. This allows the creator of the email message to be notified of subsequent updates to the issue, which can be achieved by configuring the relevant project's <a href="#">notification scheme</a> to notify the <b>Reporter</b> of updates.</p> <p>The username and email address of these newly created Jira user accounts will be the email addresses specified in the <b>From:</b> fields of these received messages. The password for these new Jira users is randomly generated and an email message is sent their addresses informing them about their new JIRA user account.</p> <p>Users created this way will be added to the default group/s of the default Jira application (and therefore take up a license for this application). See the <a href="#">Managing groups</a> documentation.  <b>Note:</b> this option is not compatible with <b>Default Reporter</b> field option below and as such, choosing the <b>Create Users</b> option will hide the <b>Default Reporter</b> option.</p>
<p><b>Default Reporter</b></p>	<p>Specify the username of a default reporter, which will be used if the email address in the <b>From:</b> field of any received messages does not match the address associated with that of an existing Jira user for example, a Jira username such as <input type="text" value="emailed-reporter"/></p> <p> <b>Note:</b></p>

	<ul style="list-style-type: none"> <li>• This option is not available if the <b>Create Users</b> checkbox is selected.</li> <li>• Please ensure that the user specified in this field has the <b>Create Issues project permission</b> for the relevant <b>Project</b> (specified above) as well as the <b>Create Comments</b> project permission for the other relevant projects to which this mail handler should add comments.</li> <li>• When an issue is created and this option is specified, the email message's <b>From:</b> field address is appended in a brief message at the end of the issue's <b>Description</b> field, so that the sender can be identified.</li> </ul>
<b>Notify Users</b>	<p>Clear this checkbox if you do not want Jira to send out an email message notifying users whose accounts have been created by the <b>Create Users</b> option above.</p> <p><b>Note:</b> this option only functions if the <b>Create Users</b> checkbox has been selected.</p>
<b>CC Assignee</b>	<p>Select this checkbox if you want Jira to automatically assign the issue created to a Jira user:</p> <ul style="list-style-type: none"> <li>• Whose email address (registered with their Jira account) matches the first matching address encountered in the <b>To:</b>, then <b>Cc:</b> and then <b>Bcc:</b> field of the email message received.</li> <li>• Who also has the <b>Assignable User project permission</b> for the relevant <b>Project</b> (specified above).</li> </ul>
<b>CC Watchers</b>	<p>Select this checkbox if you want Jira to automatically add Jira users to the issue created, where those users' email addresses (registered with their Jira accounts) match addresses encountered in the <b>To:</b>, <b>Cc:</b> or <b>Bcc:</b> fields of the email message received.</p> <p><b>Note:</b> Please note that when an issue is created, new Jira users created by the <b>Create Users</b> option (above) <i>cannot also be added</i> to the issue's watchers list by this <b>CC Watchers</b> option. Jira users must <i>already</i> exist in Jira's userbase, and must have an email address.</p>

3. Test and save your mail handler (above).

### Add a comment from the non quoted email body

**i** This email handler doesn't use regex. We advise using the **Add a comment before a specified marker or separator in the email body** handler instead. For more information, see [Set up Jira notifications](#).

This message handler creates a comment, but only uses the 'non quoted' lines of the body of the email message. A quoted line is any line that starts with a '>' or '|' symbol and such lines of text will not be added to the comment. The issue to which the comment is added is chosen from the first issue key found in the email subject. The author of the comment is taken from the address of the email message's **From:** field.

### To configure an 'Add a comment from the non quoted email body' mail handler:

1. If you have not already done so, begin configuring your mail handler (above).
2. On the **Add a comment from the non quoted email body** dialog box, complete the following fields/options:

<b>Catch Email Address</b>	<p>If specified, only email messages whose <b>To:</b>, <b>Cc:</b>, <b>Bcc:</b> lines contain the recipient specified in this field will be processed for example, <code>issues@mycompany.com</code></p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>i</b> Make sure <b>not</b> to put the Catch email address in the Bcc header when sending email but rather place it in the <b>To:</b> and <b>Cc:</b> headers. This is because mail servers tend to strip the Bcc header. Additionally, the Mail handler will look for the Catch email address in this header <b>only</b> if the header is available in the email source.</p> </div> <p>Upon specifying an address here, all email messages whose <b>To:</b>, <b>Cc:</b>, <b>Bcc:</b> lines contain addresses other than the <b>Catch Email Address</b> are ignored. This is useful if you have</p>
----------------------------	--

	<p>multiple aliases for the same mail account (e.g. <code>foo-support@example-co.com</code> and <code>bar-support@example-co.com</code> aliases for <code>support@example-co.com</code>) for multiple mail services (e.g. each one to create issues in separate Jira projects).</p> <p><b>Note:</b> in practice, <b>this option is rarely useful</b> and should not be confused with the more common <b>Default Reporter</b>. You can only specify one catch email address and one issue type per mail handler.</p>
<b>Bulk</b>	<p>This option only affects 'bulk' email messages whose header has either its <b>Precedence:</b> field set to <b>bulk</b> or its <b>Auto-Submitted</b> field not set to <b>no</b>. Such messages would typically be sent by an automated service. When such an email message is received, the following action will be performed, based on the option you choose:</p> <ol style="list-style-type: none"> <li>Ignore the email and do nothing.</li> <li>Forward the email (i.e. to the address set in the <b>Forward Email</b> text field).</li> <li>Delete the email permanently.</li> </ol>
<b>Forward Email</b>	<p>If specified, then if this mail service is unable to handle an email message it receives, an email message indicating this problem will be forwarded to the email address specified in this field. <b>Note:</b> An <a href="#">SMTP mail server</a> must be configured for this option to function correctly.</p>
<b>Create Users</b>	<p>Select this checkbox if you want Jira to create new user accounts from any received email messages whose <b>From:</b> field contains an address that does not match one associated with an existing Jira user account. This allows the creator of the email message to be notified of subsequent updates to the issue, which can be achieved by configuring the relevant project's <a href="#">notification scheme</a> to notify the <b>Reporter</b> of updates.</p> <p>The username and email address of these newly created Jira user accounts will be the email address specified in the <b>From:</b> field of the message. The password for the new user is randomly generated, and an email is sent to the new user informing them about their new account in Jira.</p> <p>Users created this way will be added to the default group/s of the default Jira application (and therefore take up a license for this application). See the <a href="#">Managing groups</a> documentation. <b>Note:</b> this option is not compatible with <b>Default Reporter</b> field option below and as such, choosing the <b>Create Users</b> option will hide the <b>Default Reporter</b> option.</p>
<b>Default Reporter</b>	<p>Specify the username of a default reporter, which will be used if the email address in the <b>From:</b> field of any received messages does not match the address associated with that of an existing Jira user for example, a Jira username such as <code>emailed-reporter</code></p> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li>This option is not available if the <b>Create Users</b> checkbox is selected.</li> <li>Please ensure that the user specified in this field has the <b>Create Issues project permission</b> for the relevant <b>Project</b> (specified above) as well as the <b>Create Comments project permission</b> for the other relevant projects to which this mail handler should add comments.</li> </ul>
<b>Notify Users</b>	<p>Clear this checkbox if you do not want Jira to send out an email message notifying users whose accounts have been created by the <b>Create Users</b> option above.</p> <p><b>Note:</b> this option only functions if the <b>Create Users</b> checkbox has been selected.</p>

3. Test and save your mail handler ([above](#)).

### Add a comment with the entire email body

This message handler creates a comment based on the entire body of the email message received. The issue to which the comment is added is chosen from the first issue key found in the email subject. The author of the comment is taken from the address of the email message's **From:** field.

### To configure an 'Add a comment with the email body' mail handler:

1. If you have not already done so, begin configuring your mail handler ([above](#)).
2. On the **Add a comment with the entire email body** dialog box, complete the following fields/options:

<b>Catch Email Address</b>	<p>If specified, only email messages whose <b>To:</b>, <b>Cc:</b>, <b>Bcc:</b> lines contain the recipient specified in this field will be processed for example, <code>issues@mycompany.com</code></p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p><b>i</b> Make sure <b>not</b> to put the Catch email address in the Bcc header when sending email but rather place it in the <b>To:</b> and <b>Cc:</b> headers. This is because mail servers tend to strip the Bcc header. Additionally, the Mail handler will look for the Catch email address in this header <b>only</b> if the header is available in the email source.</p> </div> <p>Upon specifying an address here, all email messages whose <b>To:</b>, <b>Cc:</b>, <b>Bcc:</b> lines contain addresses other than the <b>Catch Email Address</b> are ignored. This is useful if you have multiple aliases for the same mail account (e.g. <code>foo-support@example-co.com</code> and <code>bar-support@example-co.com</code> aliases for <code>support@example-co.com</code>) for multiple mail services (e.g. each one to create issues in separate Jira projects).</p> <p><b>i</b> <b>Note:</b> in practice, <b>this option is rarely useful</b> and should not be confused with the more common <b>Default Reporter</b>. You can only specify one catch email address and one issue type per mail handler.</p>
<b>Bulk</b>	<p>This option only affects 'bulk' email messages whose header has either its <b>Precedence:</b> field set to <b>bulk</b> or its <b>Auto-Submitted</b> field not set to <b>no</b>. Such messages would typically be sent by an automated service. When such an email message is received, the following action will be performed, based on the option you choose:</p> <ol style="list-style-type: none"> <li>a. Ignore the email and do nothing.</li> <li>b. Forward the email (i.e. to the address set in the <b>Forward Email</b> text field).</li> <li>c. Delete the email permanently.</li> </ol>
<b>Forward Email</b>	<p>If specified, then if this mail service is unable to handle an email message it receives, an email message indicating this problem will be forwarded to the email address specified in this field. <b>i</b> <b>Note:</b> An <a href="#">SMTP mail server</a> must be configured for this option to function correctly.</p>
<b>Create Users</b>	<p>Select this checkbox if you want Jira to create new user accounts from any received email messages whose <b>From:</b> field contains an address that does not match one associated with an existing Jira user account. This allows the creator of the email message to be notified of subsequent updates to the issue, which can be achieved by configuring the relevant project's <a href="#">notification scheme</a> to notify the <b>Reporter</b> of updates.</p> <p>The username and email address of these newly created Jira user accounts will be the email address specified in the <b>From:</b> field of the message. The password for the new user is randomly generated, and an email is sent to the new user informing them about their new account in Jira.</p> <p>Users created this way will be added to the default group/s of the default Jira application (and therefore take up a license for this application). See the <a href="#">Managing groups</a> documentation. <b>i</b> <b>Note:</b> this option is not compatible with <b>Default Reporter</b> field option below and as such, choosing the <b>Create Users</b> option will hide the <b>Default Reporter</b> option.</p>
<b>Defa</b>	

<b>Default Reporter</b>	<p>Specify the username of a default reporter, which will be used if the email address in the <b>From:</b> field of any received messages does not match the address associated with that of an existing Jira user for example, a Jira username such as <code>emailed-reporter</code></p> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li>This option is not available if the <b>Create Users</b> checkbox is selected.</li> <li>Please ensure that the user specified in this field has the <b>Create Issues project permission</b> for the relevant <b>Project</b> (specified above) as well as the <b>Create Comments project permission</b> for the other relevant projects to which this mail handler should add comments.</li> </ul>
<b>Notify Users</b>	<p>Clear this checkbox if you do not want Jira to send out an email message notifying users whose accounts have been created by the <b>Create Users</b> option above.</p> <p><b>Note:</b> this option only functions if the <b>Create Users</b> checkbox has been selected.</p>

3. Test and save your mail handler ([above](#)).

### Create a new issue from each email message

This message handler creates a new issue for each incoming message.

#### To configure an 'Create a new issue from each email message' mail handler:

- If you have not already done so, begin configuring your mail handler ([above](#)).
- On the **Create a new issue from each email message** dialog box, complete the following fields/options:

<b>Project</b>	<p>Specify the project key of the default project to which new issues are created by this handler for example, JRA.</p> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li>This field is only relevant for issue creation, not for issue commenting.</li> <li>If an email message contains an issue key in its subject line and that issue key exists in your Jira installation, the handler will add the email message content as a comment on the issue, regardless of which project the issue is in.</li> </ul>
<b>Issue Type</b>	<p>Choose the default issue type for new issues.</p>
<b>Catch Email Address</b>	<p>If specified, only email messages whose <b>To:</b>, <b>Cc:</b>, <b>Bcc:</b> lines contain the recipient specified in this field will be processed for example, <code>issues@mycompany.com</code></p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>i</b> Make sure <b>not</b> to put the Catch email address in the Bcc header when sending email but rather place it in the <b>To:</b> and <b>Cc:</b> headers. This is because mail servers tend to strip the Bcc header. Additionally, the Mail handler will look for the Catch email address in this header <b>only</b> if the header is available in the email source.</p> </div> <p>Upon specifying an address here, all email messages whose <b>To:</b>, <b>Cc:</b>, <b>Bcc:</b> lines contain addresses other than the <b>Catch Email Address</b> are ignored. This is useful if you have multiple aliases for the same mail account (e.g. <code>foo-support@example-co.com</code> and <code>bar-support@example-co.com</code> aliases for <code>support@example-co.com</code>) for multiple mail services (e.g. each one to create issues in separate Jira projects).</p> <p><b>Note:</b> in practice, <b>this option is rarely useful</b> and should not be confused with the more common <b>Default Reporter</b>. You can only specify one catch email address and one issue type per mail handler.</p>

<b>Bulk</b>	<p>This option only affects 'bulk' email messages whose header has either its <b>Precedence:</b> field set to <b>bulk</b> or its <b>Auto-Submitted</b> field not set to <b>no</b>. Such messages would typically be sent by an automated service. When such an email message is received, the following action will be performed, based on the option you choose:</p> <ol style="list-style-type: none"> <li>Ignore the email and do nothing.</li> <li>Forward the email (i.e. to the address set in the <b>Forward Email</b> text field).</li> <li>Delete the email permanently.</li> </ol>
<b>Forward Email</b>	<p>If specified, then if this mail service is unable to handle an email message it receives, an email message indicating this problem will be forwarded to the email address specified in this field. <b>Note:</b> An <a href="#">SMTP mail server</a> must be configured for this option to function correctly.</p>
<b>Create Users</b>	<p>Select this checkbox if you want Jira to create new user accounts from any received email messages whose <b>From:</b> field contains an address that does not match one associated with an existing Jira user account. This allows the creator of the email message to be notified of subsequent updates to the issue, which can be achieved by configuring the relevant project's <a href="#">notification scheme</a> to notify the <b>Reporter</b> of updates.</p> <p>The username and email address of these newly created Jira user accounts will be the email address specified in the <b>From:</b> field of the message. The password for the new user is randomly generated, and an email is sent to the new user informing them about their new account in Jira.</p> <p>Users created this way will be added to the default group/s of the default Jira application (and therefore take up a license for this application). See the <a href="#">Managing groups</a> documentation. <b>Note:</b> this option is not compatible with <b>Default Reporter</b> field option below and as such, choosing the <b>Create Users</b> option will hide the <b>Default Reporter</b> option.</p>
<b>Default Reporter</b>	<p>Specify the username of a default reporter, which will be used if the email address in the <b>From:</b> field of any received messages does not match the address associated with that of an existing Jira user for example, a Jira username such as <input type="text" value="emailed-reporter"/></p> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li>This option is not available if the <b>Create Users</b> checkbox is selected.</li> <li>Please ensure that the user specified in this field has the <b>Create Issues project permission</b> for the relevant <b>Project</b> (specified above) as well as the <b>Create Comments project permission</b> for the other relevant projects to which this mail handler should add comments.</li> <li>When an issue is created and this option is specified, the email message's <b>From:</b> field address is appended in a brief message at the end of the issue's <b>Description</b> field, so that the sender can be identified.</li> </ul>
<b>Notify Users</b>	<p>Clear this checkbox if you do not want Jira to send out an email message notifying users whose accounts have been created by the <b>Create Users</b> option above.</p> <p><b>Note:</b> this option only functions if the <b>Create Users</b> checkbox has been selected.</p>
<b>CC Assignee</b>	<p>Select this checkbox if you want Jira to automatically assign the issue created to a Jira user:</p> <ul style="list-style-type: none"> <li>Whose email address (registered with their Jira account) matches the first matching address encountered in the <b>To:</b>, then <b>Cc:</b> and then <b>Bcc:</b> field of the email message received.</li> <li>Who also has the <b>Assignable User project permission</b> for the relevant <b>Project</b> (specified above).</li> </ul>
<b>CC</b>	

<b>Watchers</b>	<p>Select this checkbox if you want Jira to automatically add Jira users to the issue created, where those users' email addresses (registered with their Jira accounts) match addresses encountered in the <b>To:</b>, <b>Cc:</b> or <b>Bcc:</b> fields of the email message received.</p> <p> Please note that when an issue is created, new Jira users created by the <b>Create Users</b> option (above) <i>cannot also be added</i> to the issue's watchers list by this <b>CC Watchers</b> option. Jira users must <i>already</i> exist in Jira's userbase, and must have an email address.</p>
-----------------	--

3. Test and save your mail handler ([above](#)).

### Add a comment before a specified marker or separator in the email body

This message handler creates a comment from the body of an email message - but ignores any part of the body past a marker or separator that matches a specified regular expression (regex).

For mail systems like Lotus Notes and Outlook, the core content of an email message is separated from other (e.g. replied or forwarded) content in the body by some predictable text string like '---- Original Message ----' or 'Extranet\n email.address/DOM/REG/CONT/CORP@CORPMAIL'. Hence, use this message handler, which can take any valid regex, to filter core from extraneous content from various different mail systems.

Also note that the issue to which the comment is added is chosen from the first issue key found in the email subject.

The **Add a comment before a specified marker or separator in the email body** mail handler has the following behavior with respect to received email messages:

- If the regex pattern (specified in the mail handler) is found, the text in the email message body before the first regex pattern match is used for the comment and the remainder of the body is discarded.
- If the regex pattern (specified in the mail handler) is not found, the entire text in the email message body is used for the comment.
- If no regex pattern is specified in the mail handler, the entire text in the email message body is used for the comment.
- If the regex expression specified in the mail handler is erroneous, the entire text in the email message body is used for the comment.

### To configure an 'Add a comment before a specified marker or separator in the email body' mail handler:

1. If you have not already done so, begin configuring your mail handler ([above](#)).
2. On the **Add a comment before a specified marker or separator in the email body** dialog box, complete the following fields/options:

<b>Split Regex</b>	<p>Specify a regular expression matching the text that separates the content of the email message mail body from other (replied or forwarded) content in the body.</p> <p> <b>Please Note:</b></p> <ul style="list-style-type: none"> <li>• The regex must begin and end with a delimiter character, typically '/'.</li> <li>• Commas are not allowed in a regex, as they are used to separate each mail handler field /option when they are integrated into a Jira service and there is not (as yet) an escape syntax.</li> </ul> <p>For example:</p> <pre>/----\s*Original Message\s*----/</pre> <p>or</p> <pre>/_____*/</pre>
<b>Cache</b>	<p>If specified, only email messages whose <b>To:</b>, <b>Cc:</b>, <b>Bcc:</b> lines contain the recipient specified in this field will be processed for example, <a href="mailto:issues@mycompany.com">issues@mycompany.com</a></p> <p></p>

<p><b>mail Address</b></p>	<p>Make sure <b>not</b> to put the Catch email address in the Bcc header when sending email but rather place it in the <b>To:</b> and <b>Cc:</b> headers. This is because mail servers tend to strip the Bcc header. Additionally, the Mail handler will look for the Catch email address in this header <b>only</b> if the header is available in the email source.</p> <p>Upon specifying an address here, all email messages whose <b>To:</b>, <b>Cc:</b>, <b>Bcc:</b> lines contain addresses other than the <b>Catch Email Address</b> are ignored. This is useful if you have multiple aliases for the same mail account (e.g. <a href="mailto:foo-support@example-co.com">foo-support@example-co.com</a> and <a href="mailto:bar-support@example-co.com">bar-support@example-co.com</a> aliases for <a href="mailto:support@example-co.com">support@example-co.com</a> ) for multiple mail services (e.g. each one to create issues in separate Jira projects).</p> <p><b>Note:</b> In practice, <b>this option is rarely useful</b> and should not be confused with the more common <b>Default Reporter</b>. You can only specify one catch email address and one issue type per mail handler.</p>
<p><b>Bulk</b></p>	<p>This option only affects 'bulk' email messages whose header has either its <b>Precedence:</b> field set to <b>bulk</b> or its <b>Auto-Submitted</b> field not set to <b>no</b>. Such messages would typically be sent by an automated service. When such an email message is received, the following action will be performed, based on the option you choose:</p> <ol style="list-style-type: none"> <li>Ignore the email and do nothing.</li> <li>Forward the email (i.e. to the address set in the <b>Forward Email</b> text field).</li> <li>Delete the email permanently.</li> </ol>
<p><b>Forward Email</b></p>	<p>If specified, then if this mail service is unable to handle an email message it receives, an email message indicating this problem will be forwarded to the email address specified in this field. <b>Note:</b> An <a href="#">SMTP mail server</a> must be configured for this option to function correctly.</p>
<p><b>Create Users</b></p>	<p>Select this checkbox if you want Jira to create new user accounts from any received email messages whose <b>From:</b> field contains an address that does not match one associated with an existing Jira user account. This allows the creator of the email message to be notified of subsequent updates to the issue, which can be achieved by configuring the relevant project's <a href="#">notification scheme</a> to notify the <b>Reporter</b> of updates.</p> <p>The username and email address of these newly created Jira user accounts will be the email address specified in the <b>From:</b> field of the message. The password for the new user is randomly generated, and an email is sent to the new user informing them about their new account in Jira.</p> <p>Users created this way will be added to the default group/s of the default Jira application (and therefore take up a license for this application). See the <a href="#">Managing groups</a> documentation. <b>Note:</b> this option is not compatible with <b>Default Reporter</b> field option below and as such, choosing the <b>Create Users</b> option will hide the <b>Default Reporter</b> option.</p>
<p><b>Default Reporter</b></p>	<p>Specify the username of a default reporter, which will be used if the email address in the <b>From:</b> field of any received messages does not match the address associated with that of an existing Jira user for example, a Jira username such as <code>emailed-reporter</code></p> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li>This option is not available if the <b>Create Users</b> checkbox is selected.</li> <li>Please ensure that the user specified in this field has the <b>Create Issues</b> <a href="#">project permission</a> for the relevant <b>Project</b> (specified above) as well as the <b>Create Comments</b> project permission for the other relevant projects to which this mail handler should add comments.</li> </ul>
<p><b>Notify</b></p>	<p>Clear this check box if you do not want Jira to send out an email message notifying users whose accounts have been created by the <b>Create Users</b> option above.</p>

**Users** ⓘ **Note:** this option only functions if the **Create Users** check box has been selected.

3. Test and save your mail handler ([above](#)).

## Custom mail handlers

You can design your own message handlers to better integrate your own processes into Jira. Such custom mail handlers configured using the standard procedure [above](#).

For more information about creating custom mail handlers, see the [Message Handler Plugin Module](#) documentation.

## Issue/comment creation

The following points describe how Jira processes each incoming email message and determines how its content gets added as either a comment to an existing issue or a new issue altogether.

- The **subject** of an email message is examined for an existing issue key:
  - If an issue key is found in the **subject**, the content of the email message's **body** is processed and added as a comment to the issue with that issue key.
  - If an issue key is NOT found in the **subject**, the **in-reply-to header** is examined:
    - If the email message is found to be a reply to another email message from which an issue was previously created, the **body** is processed and added as a comment to that issue.
    - If the email message is NOT found to be a reply, a new issue is created.

For example, an email message to a mail account `foo@example-co.com` on a POP or IMAP mail server configured against a Jira server will be processed as follows:

- Issue Creation:
  - The **subject** of the email message will become the issue summary.
    - ⚠ Since all issues require a summary, each email message intended for issue creation should include a **subject**.
  - The **body** of the email message will be the issue description.
  - A bug will be created for project 'JIRA' with the above information. (This is essentially based on the mail handler configuration [above](#)).
  - Any attachments to the email message will become attachments to the issue (assuming [attachments](#) have been enabled in Jira).
    - ⓘ To ensure compatibility with various operating systems, any of the following characters in the filename will be replaced with an underscore character: \, /, ", %, :, \$, ?, \*, <, |, >.
  - If the incoming email is set to a high priority, the corresponding issue will be created with a higher priority than the default priority that is set in your Jira system.
- Comment Creation:
  - The **body** of the email will become a comment on the issue.
  - Any attachments to the email will become attachments to the issue (assuming attachments have been enabled in Jira).

## Handy tips with mail handlers

### To allow Jira to handle email messages sent from people without a Jira user account:

1. Create an 'anonymous'/'dummy' mail account on your mail server/service ([above](#)).
2. Create an equivalent 'anonymous'/'dummy' Jira user account, whose **Email** field matches the mail account you created in the previous step.
3. When configuring your mail handler(s) ([above](#)) to handle messages from this mail account, set the **Default Reporter** to this 'anonymous'/'dummy' Jira user account.

## Best practices (pre-processing Jira email messages)

For Jira production servers, we recommend that setting up the following email message pre-processing:

- Since Jira mail handlers remove successfully processed email messages from your mail server, ensure that your mail is sent to a backup folder so that a record of what mail Jira processed is available.
- If your mail folder contains replies to Jira's email notifications, set up rules that filter out auto-replies and bounces.

If you do not do this, there is a strong possibility of mail loops between Jira and autoresponders like 'out of office' notifications. Jira sets a 'Precedence:bulk' header (unless you have disabled this) and an 'Auto-Submitted' header on outgoing email, but some autoresponders ignore it.

There is no bulletproof way of detecting whether an email is a bounce or autoreply. The following rules (in procmail format) will detect most autoreplies:

```

^From:.*mailer-daemon@
^Auto-Submitted:.auto-
^Content-Type:\ multipart/report;\ report-type=delivery-status
^Subject:\ Delivery\ Status\ Notification
^Subject:\ Undeliverable
^Subject: Returned Mail:
^From:\ System\ Administrator
^Precedence:\ auto_reply
^Subject:.*autoreply
^Subject:.*Account\ signup

```

Even with these rules, you may encounter autoreplies with nothing in the headers to distinguish it from a regular mail, In these cases you will just need to manually update the filters to exclude that sender.

- Set up a filter to catch email with huge attachments. Jira uses the standard JavaMail library to parse email, and it quickly runs out of memory on large attachments (e.g. > 50 MB given 512 MB heap). As the un-handled mail is not deleted, it will be reprocessed (causing another OutOfMemoryError) each time the mail service runs. In practice this problem is rarely seen, because most mail servers are configured to not accept email with huge attachments. Unless you are sure your mail server will not pass a huge attachment on to Jira, it is best to configure a filter to prevent Jira encountering any huge attachments.
- Set up spam filtering rules, so Jira does not have to process (and possibly create issues from) spam.

## Troubleshooting

Jira's **Logging & Profiling** page has configuration options for Outgoing and Incoming mail. Whenever you create a new (or edit an existing) mail handler (above), a **Test** button is available to allow you to test your mail handler's configuration to ensure it works as expected. A useful tip for debugging mail-related problems in Jira is to [set](#) the `-Dmail.debug=true` property on startup. This will cause protocol-level details of Jira's email interactions to be logged in `catalogina.out` (or standard output).

### Common problems

- If Jira does not appear to be creating sending emails or creating issues and comments from email, your Jira instance could be experiencing **OutOfMemory errors**. Please check your log files for OutOfMemory errors. If there are OutOfMemory errors, please restart Jira and [investigate the errors](#).
- If you find some incoming emails simply disappear, check that you have not accidentally **started a second copy of Jira** (e.g. in a staging environment) which is downloading and deleting mails. See [Disable email sending/receiving](#) for flags you should set to prevent mail being processed.
- If replies by email of Jira's notifications list Jira's SMTP server rather than the configured handler POP account (ie, in Outlook's 'Reply-to' functionality), the project needs to be configured to add a 'reply-to' header in outgoing notifications. This can be configured in the project view for that particular project in Jira's Administration.
- If HTML/Rich Text formatting is not being process correctly by Jira, this is an expected behavior. The email comment handler was designed to do plain text conversion.



### Do more with Jira

To create issues and add comments directly from your inbox, check out these apps on the [Atlassian Marketplace](#):

- [Email This Issue](#): Notify stakeholders both inside and outside the company about issues right from workflow transitions
- [Outlook App for Jira](#): connect Jira issues with emails on all devices

# Configuring Jira applications to receive email from a POP or IMAP mail server

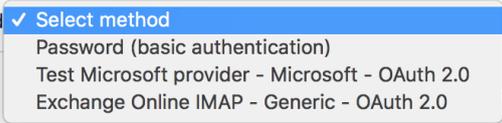
To enable Jira to [create comments and issues from email](#), you need to first configure Jira to receive email from a POP or IMAP mail server as described below.

**Note:** For all of the following procedures, you must be logged in as a user with the [Jira Administrators global permission](#).

## Add or edit a POP or IMAP mail server

1. In the upper-right corner of the screen, select **Administration**  **System**.
2. Select **Mail > Incoming Mail** to open the Incoming Mail page.
3. Click either the **Add mail server** button to define a new mail server, or the **Edit** link at the right of an existing mail server configuration, which will open the **Add/Update Mail server** page.
4. Complete the fields on this page as follows:

<b>Name</b>	Specify a short, arbitrary name to identify your mail server configuration. You could possibly just specify the email address of the POP / IMAP mail server.
<b>Description</b>	(Optional) Specify an arbitrary description that describes the POP or IMAP mail server configuration and/or what it is used for. For example, 'Email Issue Creation/Comments for <Project>'. This description appears below the <b>Name</b> of the POP / IMAP mail server in the <b>Mail servers</b> table.
<b>Service Provider</b> (not available when updating an existing POP / IMAP mail server)	<p>Choose between using your own POP / IMAP mail server (i.e. <b>Custom</b>), Gmail POP / IMAP (i.e. <b>Google Apps Mail / Gmail [POP3 / IMAP]</b>) or Yahoo! POP (i.e. <b>Yahoo! MailPlus</b>) as the service provider for your POP / IMAP mail server.</p> <p>If you use a Microsoft mail server, select <b>Custom</b>.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> If you choose any of the Gmail or Yahoo! options and then switch back to <b>Custom</b>, some of the key fields in this section will automatically be populated with the relevant POP / IMAP mail server settings for these service providers.</p> </div>
<b>Protocol</b>	<p>Choose between whether your POP / IMAP mail server is a standard (i.e. <b>POP or IMAP</b>) or a secure (i.e. <b>SECURE_POP</b> or <b>SECURE_IMAP</b>) one. If you want to use an OAuth 2.0 integration as your Authentication method, select a secure email server.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> Jira Cloud does not support self-signed certificates.</p> </div>
<b>Host Name</b>	Specify the hostname or IP address of your POP / IMAP mail server. Eg. <code>pop.yourcompany.com</code> or <code>imap.yourcompany.com</code>
<b>POP / IMAP port</b>	(Optional) The port to use to retrieve mail from your POP / IMAP account. Leave blank for default. The defaults are the following: POP: 110; SECURE_POP: 995; IMAP: 143; SECURE_IMAP: 993.
<b>Timeout</b>	(Optional) Specify the timeout period in milliseconds, which is treated as 10000 if this field is left blank. Specifying 0 or a negative value here will result in Jira waiting indefinitely for the POP / IMAP server to respond.
<b>Username</b>	The username used to authenticate your POP / IMAP account.

<p><b>Authentication method</b></p>	<p>The way to authenticate to the mail server. If you have configured your OAuth 2.0 integration, it will show on the list to select.</p> <p>Authentication method </p>
<p><b>Password</b></p>	<div data-bbox="411 398 1449 622" style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;"> <p> Google and Microsoft are planning to disable using password as an authentication method. Once this method is disabled, you will not be able to connect to your Gmail or Microsoft Exchange Online. Configure <b>OAuth 2.0</b> integration and select it as your authentication method to be able to add comments and create issues from email. Learn more about <a href="#">OAuth 2.0 integration</a>.</p> </div> <p>The password for your POP / IMAP account. You can still use it for a Yahoo or your custom email server.</p> <p>When editing an existing POP / IMAP mail server, select the <b>Change Password</b> checkbox to access and change this field.</p>

5. Click **Authorize**. You will be redirected to your service provider's site to log in to your account and authorize the connection. When you do, you will be redirected back to the application.
6. Click the **Test Connection** to check that Jira can communicate with the mail server you just configured.
7. Click **Save**.

## Add mail server

---

Use this page to add a new mail server so that Jira can

Name

A unique name to identify the mail server

Description

Service Provider

Custom

Protocol

POP



Host Name

The hostname of the mail server

POP / IMAP Port

Port Jira uses to retrieve mail

Timeout (ms)

10000

Timeout for every request

Username

The username used to connect to the mail server

Authentication method

Select method

Authorize

Test

**POP / IMAP over SSL**

You can encrypt email communications between Jira and your mail server via SSL, provided your mail server supports SSL.

Firstly, you will need to **import the mail server certificate** into a Java keystore. The process is described on the [Connecting to SSL Services](#) page.

**⚠ Important Note:** Without importing the certificate, Jira will not be able to communicate with your mail server.

# Integrating with OAuth 2.0

You can integrate your application with OAuth 2.0 authentication to connect with 3rd party apps, such as your mail server. We only support 3-legged authentication.

**i** OAuth 2.0 is only available for Jira server 8.5.9 (and further versions of 8.5.x), and 8.10 and later that use HTTPS or TLS. This is to ensure that the creation of tokens is secure.

- [Disabling Basic Authentication](#)
- [Integrating with OAuth 2.0 process for mail server](#)
- [Prerequisites](#)
- [Configuring OAuth 2.0 for Google, Microsoft, or your own custom server](#)
- [OAuth 2.0 settings details](#)
- [Troubleshooting](#)

## Disabling Basic Authentication

Some providers such as Google and Microsoft are planning on disabling Basic Authentication. When they do, you will not be able to create issues and comments from email and your connection to the Gmail and/or Microsoft Exchange Online server will no longer be operational. You do not need to update the settings in your custom email servers or other service providers if they use IMAP or POP3. They will continue to work.

**i** Currently, Jira does not support OAuth 2.0 for Microsoft Exchange Online via POP3. You can either continue using Basic Authentication until the support is provided or connect to the mail server using IMAP and then integrate with OAuth 2.0.

## Integrating with OAuth 2.0 process for mail server

You need to configure OAuth 2.0 for your Google and/or Microsoft email server and update your email server configuration. You need to be a system administrator to do that.

You need to configure the OAuth 2.0 settings first. To do that you will require specific info such as a client ID from your service provider. You can generate this data on the service providers side. Then, you need to copy the data to the OAuth plugin in your application to generate a redirect URL. You need to provide the redirect URL that your application generated at the service providers site. Once you save your configuration, you can proceed to configuring your mail server to use OAuth 2.0 as the authentication method.

## Prerequisites

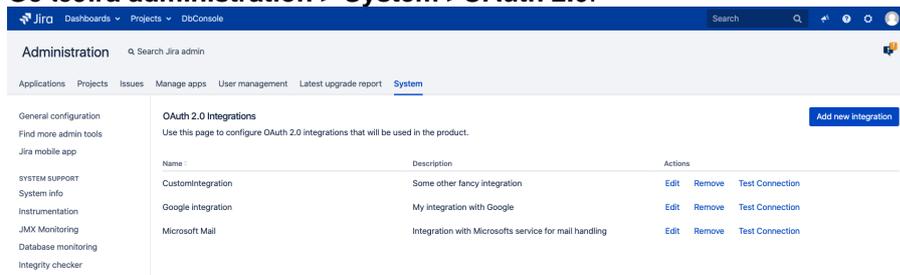
You need to ensure the following:

- [Your server needs to run over HTTPS](#). If it doesn't you will not be able to configure OAuth 2.0.
- [Your base URL needs to be configured correctly](#). This is important as the redirect URL you'll need to provide is based on the Jira's base URL.

## Configuring OAuth 2.0 for Google, Microsoft, or your own custom server

You first need to add OAuth 2.0 integration for your mail server to use. Next, you need to configure your mail server to use this integration.

## 1. Go to Jira administration > System > OAuth 2.0.



2. Click **Add new integration**.
3. Select your **Service provider**.
4. Enter your integrations name.
5. For Google and Microsoft, we will auto-fill the authorization and the token endpoint data. However, if you are using a custom service provider, you need to obtain this data from the service provider and fill it in yourself.
6. Copy the generated redirect URL, which you'll have to provide at the service providers site to obtain the client ID and client secret.

**i** If you are configuring a custom service provider, click **Generate** to receive the redirect URL.

Your redirect URL is endpoint-dependent. If you change the authorization of the token endpoint, the redirect URL needs to change as well. Click **Generate** to get a new URL.

**i** Different providers might have different requirements related to the redirect URL. For example, Google does not allow it to be a private IP address. Make sure you provide an external URL (for example of a load balancer for Data Center).

7. Go to the service provider to generate the data to enter on the plugins site to complete the integration.

Google: Go to <https://developers.google.com/identity/protocols/oauth2/web-server> to learn how to generate the required data.

Microsoft: Go to <https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-oauth2-auth-code-flow> to learn how to generate the required data.

You will need the following data for the integration:

- **Scopes**- this is the level of the authenticated user data that you allow your service provider to share with the application. For example, it can be:

Scopes \*

**i** For Google, we recommend using the `https://mail.google.com/scope` for IMAP and POP3. For Microsoft, we recommend `https://outlook.office.com/IMAP.AccessAsUser.All` or `https://outlook.office.com/POP.AccessAsUser.All, and offline_access`.

To learn more about scopes, see the detailed information at the [Microsoft](#) & [Google](#) sites.

When you complete the application registration process with your provider, you obtain the following unique credentials to authorise OAuth Client (for example, Jira) with the OAuth Server (for example, Google). Copy and paste them in Jira at the OAuth 2.0 site:

- Client ID
- Client Secret



If you use a custom service provider, you might need to generate the client ID and the client secret yourself. Make sure that the values are the same on the application and the service provider side.

8. Save your configuration.

9. On the **OAuth 2.0 integrations** page, click **Test connection** to make sure the connection works.

If you're configuring OAuth 2.0 to connect to a mail server, you can select your integration as the Authentication method for this incoming mail server. Go to **Jira configuration > System > Incoming mail** to configure your server.

For details on how to reconfigure Jira Service Management's email channels to use your OAuth 2.0 integration, see [Receiving requests by email](#).

### OAuth 2.0 settings details

Setting	Notes
Resource provider	Select <b>Google</b> or <b>Microsoft</b> if its applicable, or use <b>Custom</b> for other integrations
Name	A unique name for this integration.
Description	(Optional)
Client ID	The client ID generated by the provider. This is the public identifier of the application on the provider side.
Client Secret	The Client Secret generated by the provider. This is the shared secret between the application (such as Jira) and the provider ensuring the authorization is secure. This will not be viewable after saving.
Scopes	The required OAuth 2.0 scopes for interacting with the provider. <a href="#">Learn more about scopes</a> .
Authorization Endpoint	The HTTPS URL where authorization to use OAuth 2.0 is started.
Token Endpoint	The HTTPS URL where refresh token requests are sent. As OAuth 2.0 tokens have an expiry, Jira will periodically update the token.
Redirect URL	The redirect URL that must be saved on the provider side. This redirects the authentication flow back to Jira to complete the initial process.

### Troubleshooting

- [I fail to get an OAuth 2.0 refresh token](#)

# Jira system administration

This section of the documentation contains all the information you need to keep your Jira instance healthy and running smoothly. If you can't find the information you need, you can also check the [Jira Knowledge Base](#) and [Atlassian Answers](#) for help, and of course you can contact our legendary [Support team](#) and create an issue if you're stuck.

Search the topics in 'Jira system administration':

- System administration** [Learn more](#) about your Jira installation, like where to view your audit logs, information on Jira search indexing, and where to find your Support Entitlement Number (SEN).
- Configuring global settings** [Learn more](#) about your how to configure the settings which apply to all your users, and default settings for your Jira installation.
- Server optimization** [Learn more](#) about how to configure your Jira installation to best suit your hardware and optimize performance.

# System administration

The following section of the documentation contains details your Jira installation, like where to view the audit logs, how to find your Support Entitlement Number, and search indexing. It also contains details on backing up your instance, how to add and remove licenses, and important information on your files and directories.

- [Finding your Server ID](#)
- [Increasing Jira application memory](#)
- [Using the database integrity checker](#)
- [Precompiling JSP pages](#)
- [Logging and profiling](#)
- [Backing up data](#)
- [Restoring data](#)
- [Search indexing](#)
- [Using robots.txt to hide from search engines](#)
- [Control anonymous user access](#)
- [Licensing your Jira applications](#)
- [Viewing your system information](#)
- [Live monitoring using the JMX interface](#)
- [Monitoring database connection usage](#)
- [Monitor your instance with Jira diagnostics plugin](#)
- [Viewing Jira application instrumentation statistics](#)
- [Generating a thread dump](#)
- [Finding your Jira application Support Entitlement Number \(SEN\)](#)
- [Auditing in Jira](#)
- [Data pipeline](#)
- [Important directories and files](#)
- [Integrating Jira applications with a Web server](#)
- [Securing Jira applications with Apache HTTP Server](#)
- [Changing Jira application TCP ports](#)
- [Connecting to SSL services](#)
- [Running Jira applications over SSL or HTTPS](#)
- [Configuring security in the external environment](#)
- [Data collection policy](#)
- [Jira Admin Helper](#)
- [Raising support requests as an administrator](#)
- [Start and Stop Jira applications](#)
- [Managing LexoRank](#)
- [Jira cluster monitoring](#)
- [Scheduler administration](#)

# Finding your Server ID

The **Server ID** is an identifier for your Jira server. When creating a Jira license on [my.atlassian.com](https://my.atlassian.com), you may be prompted to enter the Server ID. You can locate your Server ID on the **System info** page.

## Finding your your Jira Server ID

1. Log in as a user with the **Jira System Administrators** [global permission](#).
2. In the upper-right corner of the screen, select **Administration**  **> System**. Select **System info** on the left menu to open the System info page.
3. The **Server ID** is displayed in the **Jira Info** section of the page.

## Jira setup wizard

If you are installing Jira for the first time, you can locate your Server ID on the **Specify your license keys** screen in the Jira setup wizard. You'll see this page if you choose to perform a custom install, or if your server is not connected to the Internet.

# Increasing Jira application memory

Java applications like Jira Software and Confluence run in a "Java virtual machine" (JVM), instead of directly within an operating system. When started, the Java virtual machine is allocated a certain amount of memory, which it makes available to Jira applications. By default, Java virtual machines are allocated 64 MB of memory, no matter how many gigabytes of memory your server may actually have available. 64 MB is inadequate for medium to large Jira application installations, and so this needs to be increased. Seeing [OutOfMemoryErrors in the logs](#) is symptomatic of this.

## On this page:

- [Step 1: Diagnosis](#)
- [Step 2: Increase available memory](#)
- [Step 3: Verify your settings](#)

### Note:

- This page addresses how to increase Heap Space memory. Confirm that you're not receiving [Perm Gen or GC Overhead errors](#).
- Make sure you do not to exceed **1024 MB** as a base configuration when installing Jira in Windows 32 bit.
- For all of the following procedures, you must be logged in as a user with the [Jira Administrators global permission](#).

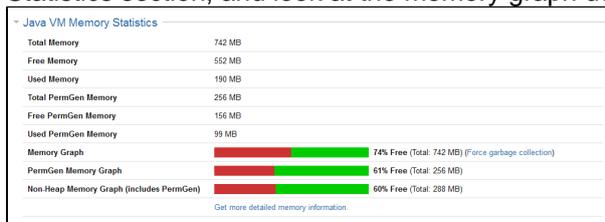
## Step 1: Diagnosis

### Assess root cause

 Often, there is a root cause for OutOfMemory Errors that may be better to address than just increasing memory. See [Jira Crashes Due to 'OutOfMemoryError Java heap space'](#) for a discussion.

### Determine Jira application usage patterns

In the upper-right corner of the screen, select **Administration**  **> System**. Select **System support > System Info** to open the System Info page. Then, scroll down the page to view the Java VM Memory Statistics section, and look at the memory graph during times of peak usage:

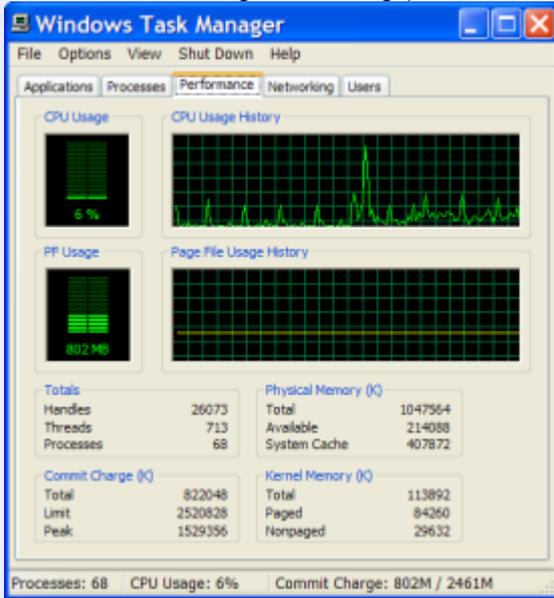


 This server has been allocated a maximum of 768 MB and a minimum of 256 MB (typically defined in the `setenv` script which is executed by running the `start-jira` script). If you are trying to see whether your settings are being picked up by Jira applications, this is where to look. Here, you can see that Jira applications have reserved 742 MB, or which 190 MB is actually in use. If this Jira application instance were running out of memory, it would have reserved the maximum available (768 MB), and would be using an amount close to this.

### Determine available system memory

**On Windows**

From the Close Programs Dialog (Press ctrl-alt-delete), select the Performance tab:



**i** The amount marked **Available** is the amount in kilobytes you have free to allocate to Jira applications. On this server, we should allocate at most 214 MB.

**On Linux**

Run `cat /proc/meminfo` to view the memory usage.

Setting the `-Xmx` above the available amount on the server runs the risk of `OutOfMemoryErrors` due to lack of physical memory. If that occurs the system will use swap space, which greatly decreases performance.

**Guidance**

As a rule of thumb, if you have fewer than 5000 issues, Jira applications should run well with the default 768 MB. Granting Jira applications too much memory can impact performance negatively, so it is best to start with 768 MB, and make modest increases as necessary. As another data point, 40,000 works well with 768 MB to 1 GB.

**Step 2: Increase available memory****Linux**

To increase heap space memory in Linux installations:

1. In your `<Jira application installation directory>/bin` (or `<Tomcat Installation Directory>/bin` for Jira WAR installations), open the `setenv.sh` file.
2. Find the sections `JVM_MINIMUM_MEMORY=` and `JVM_MAXIMUM_MEMORY=`
3. See [Diagnosis](#) above and enter the appropriate values.

**Windows (starting from .bat file)**

To configure system properties in Windows installations when starting from the .bat file:

1. In your <Jira application installation directory>/bin (or <Tomcat Installation Directory>/bin for Jira WAR installations), open the setenv.bat file.
2. Find the section **set JVM\_MINIMUM\_MEMORY=** and **set JVM\_MAXIMUM\_MEMORY=**
3. See [Diagnosis](#) above and enter the appropriate values.

### Windows service

There are two ways to configure system properties when starting [Running Jira applications as a Windows service](#), either via [command line](#) or [in the Windows registry](#).

### Setting properties for Windows services via command line

To set properties for Windows services via command line

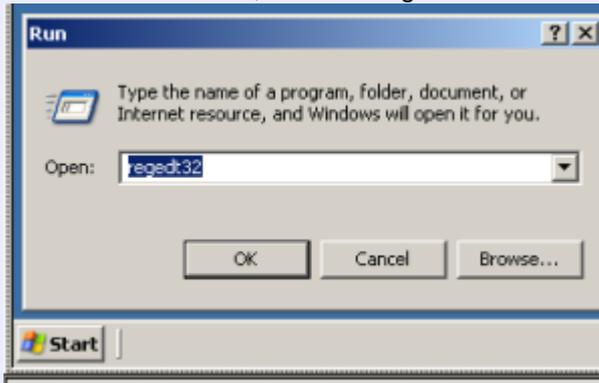


## Setting properties for Windows services via the Windows registry

In some versions of Windows, there is no option to add Java variables to the service. In these cases, you must add the properties by viewing the option list in the registry.

To set properties for Windows services via the Windows registry

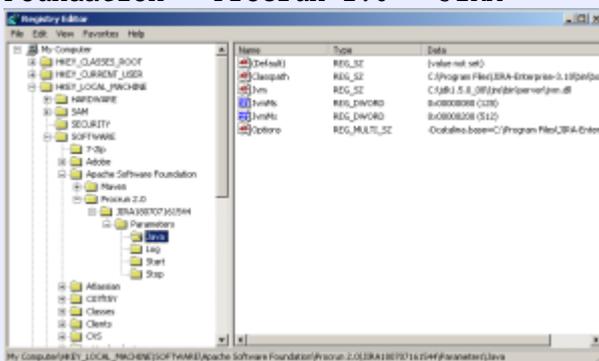
1. Go to Start > Run, and run "regedit32.exe".



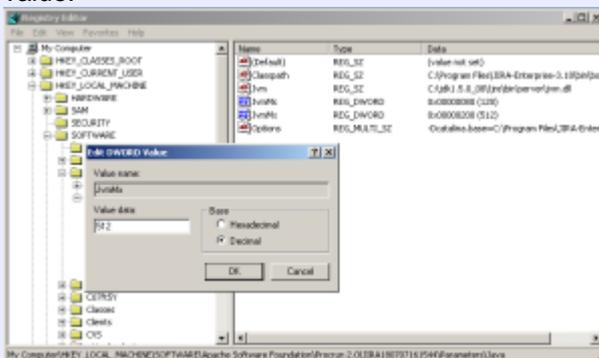
2. Find the Services entry:

**32-bit:** HKEY\_LOCAL\_MACHINE > SOFTWARE > Apache Software Foundation > Procrun 2.0 > JIRA

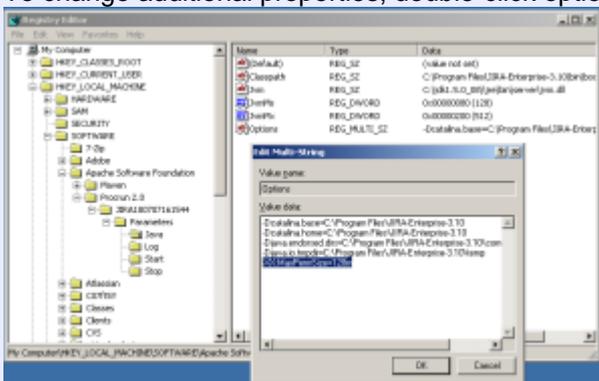
**64-bit:** HKEY\_LOCAL\_MACHINE > SOFTWARE > Wow6432Node > Apache Software Foundation > Procrun 2.0 > JIRA



3. To change existing properties, especially increasing Xmx memory, double-click the appropriate value.



4. To change additional properties, double-click options.



5. Modify the memory allocations here.

### Step 3: Verify your settings

To verify what settings are in place, check the <[Jira application home directory](#)>/logs/atlassian-jira.log or catalina.out file. A section in the startup appears like this:

```
JVM Input Arguments : -Djava.util.logging.config.file=/usr/local/jira/conf/logging.properties -XX:
MaxPermSize=256m -Xms256m -Xmx384m -Djava.awt.headless=true -Datlassian.standalone=JIRA -Dorg.apache.
jasper.runtime.BodyContentImpl.LIMIT_BUFFER=true -Dmail.mime.decodeparameters=true -Djava.util.
logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.endorsed.dirs=/usr/local/jira/endorsed -
Dcatalina.base=/usr/local/jira -Dcatalina.home=/usr/local/jira -Djava.io.tmpdir=/usr/local/jira/temp
```

**i** Look for Xmx (maximum) and Xms (minimum) settings.

This display is also available by [viewing your system information](#).

# Using the database integrity checker

Searching for common data inconsistencies, the Database Integrity Checker attempts to ensure that all Jira data is in a consistent state.

This is useful in a number of situations, e.g.

- Before migrating a project to a new workflow
- An external program is modifying Jira's database
- Troubleshooting a server crash

If an error is encountered, most of the integrity checks provide a 'repair' option which attempts to reset the data to a stable state.

 For all of the following procedures, you must be logged in as a user with the **Jira System Administrators** [global permission](#).

## Using the Integrity Checker

1. In the upper-right corner of the screen, select **Administration**  **System**.

2. Select **System support > Integrity checker** to open the Integrity Checker page.  
The integrity checker has a number of 'integrity checks' that look for common inconsistencies in Jira's stored data.

### Integrity Checker

Select one or more integrity checks from the list below to check for out of date information in the database.

- Select All**

---

- Check Issue Relations
  - Check Issue for Relation 'ParentProject'
  - Check Issue for Relation 'RelatedOSWorkflowEntry'
  - Check that all Issue Links are associated with valid issues

---

- Check Search Request
  - Check search request references a valid project

---

- Check for Duplicate Permissions
  - Check the permissions are not duplicated

---

- Check Workflow Integrity
  - Check workflow entry states are correct
  - Check workflow current step entries
  - Check JIRA issues with null status

---

- Check Field Layout Scheme Integrity
  - Check field layout schemes for references to deleted custom fields

---

- Check for invalid filter subscriptions
  - Check FilterSubscriptions for references to non-existent QuartzTriggers
  - Check FilterSubscriptions for references to non-existent SearchRequests
  - Check for existence of SimpleTriggers

---

3. Select one or more items whose data you would like to check the integrity of and click the **'Check'** button.
4. After the selected checks run, the preview screen will be shown.  
The screen provides details about the existing data inconsistencies. If any inconsistencies were found, the **'Fix'** button will also appear on the page. The messages in red describe inconsistencies that the check will correct if it is chosen and the **'Fix'** button is clicked. Messages that appear in yellow are warnings that the check will not correct; Jira will auto-recover from these inconsistencies when an action is taken on an issue.  
Select any inconsistencies that you would like to correct, then click the **'Fix'** button.  
**! Please Note:** We **strongly** recommend taking a [backup](#) of your data before correcting any data inconsistencies.
5. If any inconsistencies were found and you chose to correct them, you will be presented with a summary screen describing all the corrective actions that have taken place.

# Precompiling JSP pages

If you decided to go the extra mile and extend Jira's build process to precompile JSP pages, keep in mind that the "include" directory in the Jira web application needs to be excluded from precompilation. The reason for this is that the JSP files in the "include" directory are not proper JSP files, but are includes that are only meant to be compiled as part of larger JSP pages.

For example, to exclude the JSP pages in the "include" directory when using Maven use the <exclude> sub-element of the <ant:jspc> task, as shown:

```
<ant:path id="jspc.classpath">
  <ant:pathelement location="${tomcat.home}/common/lib/jasper-runtime.jar"/>
  <ant:pathelement location="${tomcat.home}/common/lib/jasper-compiler.jar"/>
  <ant:pathelement location="${tomcat.home}/common/lib/servlet.jar"/>
  <ant:path refid="maven-classpath"/>
  <ant:path refid="maven.dependency.classpath"/>
  <ant:pathelement path="${maven.build.dest}"/>
  <ant:pathelement path="${java.home}/lib/tools.jar"/>
</ant:path>
<ant:jspc
  package="${pom.package}.jsp"
  destDir="${jspOutDir}"
  srcdir="${warSource}"
  uriroot="${warSource}"
  uribase="/${pom.artifactId}"
  verbose="2"
  classpathref="jspc.classpath">
  <ant:include name="**/*.jsp"/>
  <ant:exclude name="**/includes/**/*.jsp"/>
</ant:jspc>
```

# Logging and profiling

## Logging

Jira uses a powerful logging module called `log4j` for runtime logging.

 For all of the following procedures, you must be logged in as a user with the **Jira System Administrators** global permission.

### On this page:

- [Logging](#)
- [Profiling](#)

### Log file location

The logs are written to the `log` subdirectory of your [Jira application home directory](#) (or elsewhere if you have configured a different location). You can view the location of the `atlassian-jira.log` in the **'File Paths'** section of the [system information](#) page.

- Security-related information (e.g. login, logout, session creation/destruction, security denials) is written to `atlassian-jira-security.log`.

### Changing the location of the log

In the `log4j.properties` file (located in the [Jira application installation directory](#)):

1. Change the following line:

```
log4j.appender.filelog=com.atlassian.jira.logging.JiraHomeAppender
```

...to this:

```
log4j.appender.filelog=org.apache.log4j.RollingFileAppender
```

2. Change the following line to point to the new location of the log file:

```
log4j.appender.filelog.File=atlassian-jira.log
```

**Note:** If you change the location of your log files, they will no longer be included when you generate a support zip. This means you'll need to attach your logs to any support requests manually.

### Logging levels

There are five logging levels available in `log4j`: `'DEBUG'`, `'INFO'`, `'WARN'`, `'ERROR'` and `'FATAL'`. Each logging level provides more logging information than the level before it:

- `'DEBUG'`
- `'INFO'`
- `'WARN'`
- `'ERROR'`
- `'FATAL'`

`'DEBUG'` provides the most verbose logging and `'FATAL'` provides the least verbose logging. The default level is `WARN`, meaning warnings and errors are displayed. Sometimes it is useful to adjust this level to see more detail.

 **Please be aware:** the `'DEBUG'` setting may cause user passwords to be logged.

The default logging levels can be changed either

- [temporarily](#) your change to the logging level will not persist after you next restart Jira, or
- [permanently](#) your change to the logging level will persist, even after you restart Jira.

For example, when troubleshooting, you might temporarily change the logging level from 'WARNING' to 'INFO' so as to get a more detailed error message or a stack trace. If you are unsure of which logging categories to adjust, the most helpful information generally comes from the `log4j.rootLogger` category and the `log4j<category>.com.atlassian` categories.

### Temporarily changing the logging level

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. Select **System support > Logging & Profiling** to open the Logging page, which lists all defined log4j categories (as package names) and their current logging levels.
3. To change logging level of a category, click linked logging level associated with the relevant package name. To turn off logging of a category, click the 'OFF' link associated with the relevant package name.

### Permanently changing the logging level

1. Edit the `log4j.properties` file (located in the [Jira application installation directory](#)).
2. Locate the section:

```
log4j.logger.com.atlassian = WARN, console, filelog
log4j.additivity.com.atlassian = false
```

and make your desired changes (e.g. change the `WARN` to `DEBUG`).

-  The `log4j.properties` file that ships with Jira has the default logging levels specified. For more information about log4j (e.g. how to define new logging categories), and about the format of the `log4j.properties` file, please refer to the documentation on the [log4j](#) site.
3. Restart Jira.

 **Please note:** If your application server configures logging itself, you may need to remove the `log4j.properties` file. You may also need to remove the entire `log4j.jar` file to get logging to work.

## Profiling

If you are experiencing performance issues with Jira, it is often helpful to see where the slow-downs occur. To do this you can enable profiling as described below, and then analyze the performance traces that Jira will produce for every request. Profiling traces report time spent in tenths of milliseconds.

Profiling information is available in the `atlassian-jira-profiler.log` file.

An example of a profiling trace is shown below:

```
[Filter: profiling] Turning filter on [jira_profile=on]
[116ms] - /secure/Dashboard.jspa
  [5ms] - IssueManager.execute()
    [5ms] - IssueManager.execute()
      [5ms] - Searching Issues
    [29ms] - IssueManager.execute()
      [29ms] - IssueManager.execute()
        [29ms] - Searching Issues
          [28ms] - Lucene Query
            [23ms] - Lucene Search
```

Profiling can be enabled either

- **temporarily** profiling will be enabled until you next restart Jira, or
- **permanently** profiling will remain enabled, even after you restart Jira.

### Temporarily enabling profiling

1. In the upper-right corner of the screen, select **Administration**  **> System**.

2. Select **System support > Logging & Profiling** to open the Logging page, which lists all defined log4j categories (as package names) and their current logging levels.
3. Scroll to the **'Profiling'** section at the end of the page. This section will inform you whether profiling is currently turned 'ON' or 'OFF' and will provide you with 'Disable' or 'Enable' profiling links respectively.
  - To turn Profiling 'ON', click the **'Enable'** profiling link. Jira will start generating profiling traces in the `atlassian-jira-profiler.log` file.
  - To turn Profiling 'OFF', click the **'Disable'** profiling link.

### Permanently enabling profiling

1. In your Jira installation directory, edit the `atlassian-jira/WEB-INF/web.xml` file.
2. Find the following entry:

```
<filter>
  <filter-name>profiling</filter-name>
  <filter-class>com.atlassian.jira.web.filters.JIRAProfilingFilter</filter-class>
  <init-param>
    <!-- specify the which HTTP parameter to use to turn the filter on or off -->
    <!-- if not specified - defaults to "profile.filter" -->
    <param-name>activate.param</param-name>
    <param-value>jira_profile</param-value>
  </init-param>
  <init-param>
    <!-- specify the whether to start the filter automatically -->
    <!-- if not specified - defaults to "false" -->
    <param-name>autostart</param-name>
    <param-value>>false</param-value>
  </init-param>
</filter>
```

3. Modify the `autostart` parameter to be **true** instead of **false**. That is:

```
<init-param>
  <!-- specify the whether to start the filter automatically -->
  <!-- if not specified - defaults to "false" -->
  <param-name>autostart</param-name>
  <param-value>>true</param-value>
</init-param>
```

4. Save the file. Profiling will be enabled when you restart Jira and you will be able to find the profiling information in the `atlassian-jira-profiler.log` file.

# Logging email protocol details

To assist in resolving email issues, it can be useful to know exactly what is passing over the wire between Jira and SMTP, POP or IMAP servers. This page describes how to enable protocol-level logging.

To do this

Set **Dmail.debug=true** and restart Jira. Refer to [Setting properties and options on startup](#) for details on how to do this.

Output

In the logs, you should then see JavaMail initialize the first time a mail operation is run:

```

DEBUG: JavaMail version 1.3.2
DEBUG: java.io.FileNotFoundException: /usr/local/jdk1.6.0/jre/lib/javamail.providers (No such file or
directory)
DEBUG: !anyLoaded
DEBUG: not loading resource: /META-INF/javamail.providers
DEBUG: successfully loaded resource: /META-INF/javamail.default.providers
DEBUG: Tables of loaded providers
DEBUG: Providers Listed By Class Name: {com.sun.mail.smtp.SMTPSSLTransport=javax.mail.Provider[TRANSPORT,
smtps,com.sun.mail.smtp.SMTPSSLTransport,Sun Microsystems, Inc], com.sun.mail.smtp.SMTPTransport=javax.mail.
Provider[TRANSPORT,smtp,com.sun.mail.smtp.SMTPTransport,Sun Microsystems, Inc], com.sun.mail.imap.
IMAPSSLStore=javax.mail.Provider[STORE,imaps,com.sun.mail.imap.IMAPSSLStore,Sun Microsystems, Inc], com.sun.
mail.pop3.POP3SSLStore=javax.mail.Provider[STORE,pop3s,com.sun.mail.pop3.POP3SSLStore,Sun Microsystems,
Inc], com.sun.mail.imap.IMAPStore=javax.mail.Provider[STORE,imap,com.sun.mail.imap.IMAPStore,Sun
Microsystems, Inc], com.sun.mail.pop3.POP3Store=javax.mail.Provider[STORE,pop3,com.sun.mail.pop3.POP3Store,
Sun Microsystems, Inc]}
DEBUG: Providers Listed By Protocol: {imaps=javax.mail.Provider[STORE,imaps,com.sun.mail.imap.IMAPSSLStore,
Sun Microsystems, Inc], imap=javax.mail.Provider[STORE,imap,com.sun.mail.imap.IMAPStore,Sun Microsystems,
Inc], smtps=javax.mail.Provider[TRANSPORT,smtps,com.sun.mail.smtp.SMTPSSLTransport,Sun Microsystems, Inc],
pop3=javax.mail.Provider[STORE,pop3,com.sun.mail.pop3.POP3Store,Sun Microsystems, Inc], pop3s=javax.mail.
Provider[STORE,pop3s,com.sun.mail.pop3.POP3SSLStore,Sun Microsystems, Inc], smtp=javax.mail.Provider
[TRANSPORT,smtp,com.sun.mail.smtp.SMTPTransport,Sun Microsystems, Inc]}
DEBUG: successfully loaded resource: /META-INF/javamail.default.address.map
DEBUG: !anyLoaded
DEBUG: not loading resource: /META-INF/javamail.address.map
DEBUG: java.io.FileNotFoundException: /usr/local/jdk1.6.0/jre/lib/javamail.address.map (No such file or
directory)
DEBUG: getProvider() returning javax.mail.Provider[STORE,pop3,com.sun.mail.pop3.POP3Store,Sun Microsystems,
Inc]
DEBUG POP3: connecting to host "localhost", port 110, isSSL false
S: +OK Dovecot ready.
C: USER pop-test
S: +OK
C: PASS pop-test
[Filter: profiling] Using parameter [jira_profile]
[Filter: profiling] defaulting to off [autostart=false]
[Filter: profiling] Turning filter off [jira_profile=off]
S: +OK Logged in.
C: STAT
S: +OK 2 1339
C: NOOP
S: +OK
C: TOP 1 0
S: +OK
Return-path: <pop-test@atlassian.com>
Envelope-to: pop-test@localhost
Delivery-date: Wed, 28 Feb 2007 16:28:26 +1100
Received: from pop-test by teacup.atlassian.com with local (Exim 4.63)
(envelope-from <pop-test@atlassian.com>)
id 1HMHMY-0007gB-80
for pop-test@localhost; Wed, 28 Feb 2007 16:28:26 +1100
Date: Wed, 28 Feb 2007 16:28:26 +1100
From: Jeff Turner <jeff@atlassian.com>
To: pop-test@localhost
Subject: Testing to me - Wed Feb 28 16:28:23 EST 2007
Message-ID: <20070228052826.GA29514@atlassian.com>
MIME-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Disposition: inline
User-Agent: Mutt/1.5.13 (2006-08-11)
Lines: 0

```

## Related pages

- [Logging and profiling](#)

# Backing up data

This page describes how to back up your Jira data, and establish processes for maintaining continual backups. Backing up your Jira data is the first step in upgrading your server to a new Jira revision, or splitting your Jira instance across multiple servers. See also [Restoring data](#) and [Restoring a project from backup](#).

Creating a complete backup of Jira consists of two stages:

- 1. [Backing up database contents](#)
  - [Option 1: Using native database backup tools](#)
  - [Option 2: Using Jira's XML backup utility](#)
- 2. [Backing up the data directory](#)

## 1. Backing up database contents

There are two options available to back up database contents:

Option 1: Using native database backup tools RECOMMENDED

Option 2: Using Jira's XML backup utility

 For production use, it is **strongly recommended** that for regular backups, you use native database backup tools instead of Jira's XML backup service.

When Jira is in use, XML backups are not guaranteed to be consistent as the database may be updated during the backup process. Jira does not report any warnings or error messages when an XML backup is generated with inconsistencies and such XML backups will fail during the restore process. Native database backup tools offer a much more consistent and reliable means of storing (and restoring) data while Jira is active.

### Option 1: Using native database backup tools

All serious databases come with tools to back up and restore databases (the 'MS' in RDBMS). We strongly recommend these tools in preference to the XML backup option described below, as they:

- ensure integrity of the database by taking the backup at a single point in time
- are much faster and less resource-intensive than Jira's XML backup.
- integrate with existing backup strategies (e.g. allowing one backup run for all database-using apps).
- may allow for incremental (as opposed to 'full') backups, saving disk space.
- avoid character encoding and format issues relating to Jira's use of XML as a backup format.

See the documentation for your database on how to set up periodic backups. This typically involves a cron job or Windows scheduled task invoking a command-line tool like [mysqldump](#) or [pg\\_dump](#).

### Option 2: Using Jira's XML backup utility

 For all of the following procedures, you must be logged in as a user with the **Jira System Administrators** [global permission](#).

To perform a once-off backup, e.g. before an upgrade, follow the steps below.

 You can also configure scheduled XML backups, as described in [Automating Jira application backups](#).

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. Select **Import & Export > Backup System** to open the Backup Jira data page.  
*Screenshot: [The Backup Jira Data Page](#)*

### Backup JIRA data ?

This will backup the contents of the database in a portable XML format.

You can use this backup to move JIRA between different databases if required, as well as creating a backup that you can use if something goes wrong. To backup to a file on the server, enter the filename below.

The backup file will be placed here: `C:\jira\home\export`

**⚠** Attachments will not be backed up. This needs to be done manually.  
XML generation is complex so there might be a delay before it completes!

File name

---

- i** As shown in the screenshot above, the backup will be stored within the `export` subdirectory of the [Jira application home directory](#).
- 3. In '**File name**' field, type the name of the backup file.
- i** Ensure that Jira has the necessary file system permissions to write to this location.
- 4. Click the '**Backup**' button and wait while your Jira data is backed up.
- i** Jira will save your XML backup as a zipped archive file.
- 5. When the backup is complete, a message will be displayed, confirming that Jira has written its data to the file you specified.

## 2. Backing up the `data` directory

Its crucial that you back up your Jira applications `data` directory, which is a sub-directory of the [Jira application home directory](#) (`jira-home` for short). The `data` directory contains application data for your Jira instance. For example, issue attachments are stored in the `<jira-home>\data\attachments` directory.

### **i** Backing up the JIRA index

The Jira index is stored in a different sub-directory, `<Jira-home>\caches`. On large instances, we recommend that you enable 'restorable index' in the system options to create backups of the index that you can restore later.

There's no one specific way to back up the `data` directory, but here are a couple of methods you might consider:

- On MS Windows, a batch script copying the directory can be written and scheduled periodically (Programs > Accessories > System Tools > Scheduled Tasks).
- On Linux/Solaris, you can write a small shell script, placed in `/etc/cron.daily`, backing up files to a directory like `/var/backup/jira`. It is best to copy an existing script in `/etc/cron.daily` to ensure local conventions (file locations, lockfiles, permissions) are adhered to.

If you have put your `attachments` directory in a custom location rather than inside the `data` directory, you'll need to back up your `attachments` directory separately.

If you have a large file, refer to the following guide on [How to Transfer Large Files to Atlassian](#).

# Automating Jira application backups

Jira applications can be configured to automatically create an XML backup of Jira application data on a routine basis.

## **Please note:**

- The XML backup includes all data in the database. However, it **does not include** your [attachments directory](#), [Jira application home directory](#), or [Jira application installation directory](#), which are stored on the filesystem.
- You can also perform XML backups manually. See [Backing up data](#) for details.
- Be aware that after [installing Jira applications](#) and running the [setup wizard](#), a backup service will automatically be configured to run every 12 hours.

**!** For production use or large Jira application installations, it is **strongly recommended** that you use [native database-specific tools](#) instead of the XML backup service. XML backups are not guaranteed to be consistent, as the database may be updated during the backup process. Inconsistent backups are created successfully without any warnings or error messages, but fail during the restore process. Database-native tools offer a much more consistent and reliable means of storing data.

## To configure automated Jira application backups:

1. Log in as a user with the [Jira System Administrators global permission](#).
2. Select **Administration > System > Advanced > Services** (tab) to open the **Services** page, which lists the current services running on this system. By default, there should be at least one 'Mail Queue Service' running, which cannot be deleted.

Name / Class	Properties	Delay (mins)	
Service Provider Session Remover <small>com.atlassian.sai.jira.scheduling.JiraPluginSchedulerService</small>	<ul style="list-style-type: none"><li>• pluginJobName: Service Provider Session Remover</li><li>• repeatInterval: 28800000</li><li>• initiallyFired: true</li></ul>	480	Edit   Delete
com.atlassian.jira.plugin.ext.bamboo.service.PlanStatusUpdateServiceImpl;job <small>com.atlassian.sai.jira.scheduling.JiraPluginSchedulerService</small>	<ul style="list-style-type: none"><li>• pluginJobName: com.atlassian.jira.plugin.ext.bamboo.service.PlanStatusUpdateServiceImpl;job</li><li>• repeatInterval: 60000</li><li>• initiallyFired: true</li></ul>	1	Edit   Delete
com.atlassian.streams.internal.ActivityProviderConnectionMonitorImpl;activityProviderMonitor <small>com.atlassian.sai.jira.scheduling.JiraPluginSchedulerService</small>	<ul style="list-style-type: none"><li>• pluginJobName: com.atlassian.streams.internal.ActivityProviderConnectionMonitorImpl;activityProviderMonitor</li><li>• repeatInterval: 300000</li><li>• initiallyFired: true</li></ul>	5	Edit   Delete
NotificationCacheUpdateJob-job <small>com.atlassian.sai.jira.scheduling.JiraPluginSchedulerService</small>	<ul style="list-style-type: none"><li>• pluginJobName: NotificationCacheUpdateJob-job</li><li>• repeatInterval: 86400000</li><li>• initiallyFired: true</li></ul>	1440	Edit   Delete
Backup Service <small>com.atlassian.jira.service.services.export.ExportService</small>	<ul style="list-style-type: none"><li>• USE_DEFAULT_DIRECTORY: true</li></ul>	720	Edit   Delete
PluginLicenseExpiryJob-job <small>com.atlassian.sai.jira.scheduling.JiraPluginSchedulerService</small>	<ul style="list-style-type: none"><li>• pluginJobName: PluginLicenseExpiryJob-job</li><li>• repeatInterval: 86400000</li><li>• initiallyFired: true</li></ul>	1440	Edit   Delete
Mail Queue Service <small>com.atlassian.jira.service.services.mail.MailQueueService</small>		1	Edit

**Add Service**

Add a new service by entering a name and class below. You can then edit it to set properties. Mail handlers should be added or edited via our new [Incoming Mail](#) section.

Name:

Class:   
Built-in Services

- Debugging service
- Run Jelly script
- Backup service

Delay:   
Delay between running time, in minutes.

3. In the **Add Service** form towards the end of the page, complete the following fields:
  - **Name** a descriptive name for the backup service, such as `Backup Service`.
  - **Class** the appropriate fully-qualified class name for the **Backup service** using either of the following methods:
    - Select the **Backup service** from the list of Jira application **Built-in Services**. To do this:
      - a. Click the **Built-in Services** link below the **Class** field to expand the list of Jira application built-in service classes.

- b. Click the **Backup service** link. The **Class** field will automatically be populated with the following class text string `'com.atlassian.jira.service.services.export.ExportService'`
    - Type the fully-qualified class name `'com.atlassian.jira.service.services.export.ExportService'` into the **Class** field.
    - **Delay** enter the number of minutes between backups. A good default for this would be 720 minutes (12 hours) or 1440 minutes (24 hours).
      - 📘 **Please Note:** The interval specified in the Backup Service Delay (mins) is the time *when the next backup job will run since the last server restart*. Backup services cannot be scheduled to run at a specific time of day - please see [JRA-1865](#) for more on this.
4. Click the **Add Service** button. The **Edit Service** page is displayed.

### Edit Service: Backup Service (Long)

Enter text values for service properties below. Any empty fields will be set to NULL in the Service's initialisation.

Use Default Directory

Default directory is `[jira.home]/export/`

Date format

Optional simple date format.

Delay

Delay - in minutes

You can also adjust the delay period of this service. Note that if you adjust this delay, the service will be restarted.

5. Complete the following items on this page:
  - For the **Date format** field, specify the format which Jira applications will use to name the individual backup files. This format can be anything that [SimpleDateFormat](#) can parse. A good default is `'yyyy-MMM-dd-HHmm'`, which would generate files named like this: `'2007-Mar-05-1322'`.
  - For the **Delay** field, modify the number of minutes between backups if necessary.
  - If the **Use Default Directory** checkbox is displayed, see the [note](#) below.
6. Click the **Update** button. Your backup service is now configured. XML backups will be performed according to the schedule you specified in the **Delay** field.
  - For every successful backup, a zipped file of your XML backup will be saved in the backup directory.
  - If a scheduled backup fails for any reason, the zipped XML backup file will be saved into the 'corrupted' directory, which is directly under your nominated backup directory. A file explaining the reason for the failure will be written to the 'corrupted' directory. This file will have the same name as the backup file, but with the extension `'.failure.txt'`.
  - 📘 Jira applications will create the 'corrupted' directory if required - you do not need to create it.

#### 📘 About custom backup directories

The **Use Default Directory** checkbox (not shown in screenshot above) is for legacy Jira application installations (prior to Jira 4.2), which have backup services that use custom directories.

If you are using Jira 5.1.0 or earlier, the **Use Default Directory** will always be displayed, as the option of using custom directories has been deprecated. If you are using Jira 5.1.1 or later, the **Use Default Directory** checkbox will only be displayed if you upgraded from a version prior to 4.2 and you are editing an existing backup service which used a custom directory.

- If you are not using a legacy backup service with a custom directory, select the **Use Default Directory** checkbox. If you do not, *your backup service may not work correctly*.
- If you are using a legacy backup service with a custom directory, you can choose between using the default directory or your custom directory (cannot be edited). Note, if you choose the default directory option, you will not be able to choose the custom directory option.

The default directory location is the `export` subdirectory of the [Jira application home directory](#).

# Preventing users from accessing Jira applications during backups

 For production use, it is **strongly recommended** that for regular backups, you use [native database backup tools](#) instead of the Jira application XML backup service.

When Jira applications are in use, XML backups are not guaranteed to be consistent as the database may be updated during the backup process. Jira applications do not report any warnings or error messages when an XML backup is generated with inconsistencies and such XML backups will fail during the restore process. Native database backup tools offer a much more consistent and reliable means of storing (and restoring) data.

If you perform an XML backup (e.g. when upgrading Jira applications via a test environment or migrating Jira applications to another server), you can follow one of these methods to prevent users from accessing Jira applications and minimize inconsistencies in the backup file:

- **Recommended method:**

- If you have an Apache or other web/proxy server sitting in front of Jira applications, then you can stop Apache from proxying to Jira applications, and serve a static HTML page with a nice message along the lines of "Jira applications are undergoing maintenance". Note:
  - The administrator must be able to access Jira applications directly (not through Apache) to perform the XML backup.
  - This method does not require Jira applications to be restarted.

- **Alternative method 1:**

1. Shut down all Jira applications, configure them to listen on a different port and restart. Do this by editing the `server.xml` file. Change the following section:

```
<Connector port="8080"
    maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25" maxSpareThreads="
75" useBodyEncodingForURI="true"
    enableLookups="false" redirectPort="8443" acceptCount="100" connectionTimeout="
20000" disableUploadTimeout="true" />
```

- Note: If you have enabled HTTPS, then you would need to edit the `HTTPS Connector` section as well.
2. Restart all Jira applications and do the XML backup.
  3. Shut down all Jira applications, change all the settings back, then re-start the applications.

- **Alternative method 2:**

- If you have a firewall in front of your Jira applications, you could stop requests from getting through or change the port number that it uses. Note:
  - The administrator will need to log into your Jira applications on the temporary port number (or access it from behind the firewall), to perform the XML backup.
  - This method does not require Jira applications to be restarted.

 **Before you start:**

Whichever method you choose, we recommend setting an [announcement banner](#) to warn your users that Jira applications will be unavailable for a period of time.

# Restoring data

This process is typically conducted towards the end of [migrating Jira applications to another server](#) or [splitting Jira applications across multiple servers](#).

If you wish restore a single project from your backup into an existing Jira instance, refer to these instructions on [restoring a project from backup](#) instead.

Restoring Jira from backup is a three stage process:

1. *(Optional)* [Disable email sending/receiving](#)
2. Restore data to the database
  - Option 1:** [Restoring information from a native backup](#)
  - Option 2:** [Restoring data from XML to the database](#)
3. [Restore the attachments to the attachments directory](#) (if [attachments](#) were backed up).  
If you've performed native backup, you also need to copy/restore the `/data` folder manually to your new server.

# Restoring a project from backup

This page describes how to restore a single project from a backup file into your Jira instance. This also includes instructions on how to migrate a project from JiraCloud to Jira Server.

This feature is particularly useful if you do not wish to overwrite the existing projects or configuration of your Jira instance by importing the entire backup. Your backup file must have been created using Jira's backup tool. You cannot import a project from a backup using your native database tools.

If you wish to restore a project from a backup file into a **new empty Jira instance**, we highly recommend that you **do not use the Project Import tool**. Restoring the entire backup file into the new instance and then deleting unwanted projects is much simpler in this scenario, as you will retain the configuration settings from your backup. Instructions on moving a project to a new instance are available on the [splitting a Jira instance](#) page. Projects can be deleted via the 'Projects' page in Jira, which is accessed from the '\*Administration' menu.

## Before you begin

Restoring a project from a backup is not a trivial task. You may be required to change the configuration of your target Jira instance to accommodate the project import. Additionally, the Project Import data mapping can be resource intensive on your hardware and may take a long time to complete, if you are importing a large project.

 We strongly recommend that you perform a [full backup](#) of your target Jira instance before attempting to restore a project into it.

 **Note:** For all of the following procedures, you must be logged in as a user with the [Jira Administrators global permission](#).

## Project import restrictions

The Project Import tool will only import a project between identical instances of Jira. That is;

- The [version](#) of Jira in which your backup was created must be identical to the version of your target Jira instance, e.g. if your backup file was created in Jira 6.4, then your target instance of Jira must be version 6.4.
- If your instance of Jira had any [custom field plugin version mismatch](#) apps installed when the backup file was created, and the custom field was used in your project, then your target instance of Jira must have the same version of the apps installed for the Project Import tool to automatically work.

 Starting from Jira 7.0, the Project Import functionality between identical instances of Jira **supports some Active Objects data**. For example:

- Data that will be imported: Jira Software's sprint data and ranking data
- Data that will *not* be imported: Jira Software's board configuration data and Service Desk customer portals

For more information about extending the Project Import functionality, see [Guide - Extending the Jira Import app](#).

If any of these restrictions apply and you still wish to restore your project from backup, you will need to create a compatible backup file before importing your project by following the appropriate instructions below.

## Jira versions do not match

- If your backup file was created in an earlier version of Jira than your target instance of Jira:

### On this page:

- [Before you begin](#)
- [Restoring a project from Jira Cloud to Jira Server](#)
- [Restoring your project](#)

1. Set up a test Jira instance, which is the same version as your target instance of Jira. Make sure that the test Jira instance uses a separate database and index from your target Jira instance.
  2. [Import the backup file](#) into a test Jira instance. (This will completely overwrite the test instance.)
  3. [Create a new backup file](#) from your test Jira instance. You can now use this backup to import a specific project into your target production instance.
- If your backup file is from a later version of Jira than your target instance of Jira:
    1. [Upgrade](#) the version of your target instance of Jira to match the version of Jira in which the backup was created.

#### Custom fields app versions do not match

- If the custom fields app from your backup is an earlier version than the custom fields app in your target instance of Jira:
  1. [Import the backup file](#) into a test Jira instance. Make sure that the test Jira instance uses a separate database and index from your target Jira instance, as the import will overwrite all data in the database.
  2. In your test Jira instance, upgrade your version of your custom fields app to match the version of the app in your target instance of Jira.
  3. [Create a new backup file](#) from your test Jira instance.
- If the custom fields app from your backup is a later version than the custom fields app in your target instance of Jira:
  1. Upgrade the custom fields app version of your target instance of Jira to match the version of Jira in which the backup was created.

#### Restoring a project from Jira Cloud to Jira Server

You cannot import a project directly from JiraCloud to Jira Server the importer will display errors about version mismatches. If you want to restore a project from JiraCloud to Jira Server, follow the steps below:

1. Install a new Jira instance (in addition to the one that you want to import your project into). This will be a temporary instance that is used to store a full Jira import from JiraCloud. Ensure that the version of this temporary instance matches the version of the Jira instance that you want to import your project into, e.g. JIRA 6.2.
2. Do a full Jira migration from JiraCloud to the temporary Jira instance. See [Migrating from Jira Cloud to Jira Server applications](#).
3. Export the desired project from the temporary Jira instance.
4. Import the project into your desired Jira instance, by following the instructions in the [Restoring your project](#) section below.
5. (optional) Delete the temporary Jira instance, once the project has completed.

#### Restoring your project

The Project Import tool will attempt to map the data in your backup file into your target Jira instance. If the project you are restoring does not exist in your target Jira instance, it will create and populate the project with data from your backup. If the project already exists and is empty, it will attempt to populate the data from your backup into the project.

##### Why should I create an empty project in my target JIRA instance?

It is important to note that the primary task of the Project Import tool is to restore the data from your backup project into your target Jira instance. While the Project Import tool can create a project if one does not exist in your target Jira instance, it does not recreate any configuration settings that affect the data (e.g. screen schemes). If you wish to retain any configuration settings from your original project, we recommend that you create an empty project in your target instance with the necessary configuration settings before importing the data from your backup project.

You may wish to carry out the following setup tasks to ensure that your target Jira instance is prepared to receive a project import beforehand. This can improve the time taken to validate the data mappings to your target Jira instance.

If you are confident that your Jira instance is set up appropriately, you can skip straight to the [Project Import tool](#) instructions. If there are any problems mapping the data from your backup file to your target Jira instance, the Project Import tool will present validation errors for you to address.

## Preparing your target Jira instance

The Project Import tool does not automatically add missing project entities (e.g. user groups, issue priorities, custom field types) or fix incorrect associations (e.g. issue types in workflow schemes), so some manual work is required to set up your target Jira instance so that your project can be restored. If the Project Import wizard cannot find a valid target location for any of the backup project data, it will not be able to restore the project. The instructions below describe the setup activities that address the most common data mapping problems that occur when restoring a project from a backup.

We recommend that you perform as much of the configuration of your target Jira instance as possible, prior to starting the project import. However, if you do not have the information available to complete these setup activities beforehand, the Project Import wizard will inform you of any problems that need your attention. Alternatively, you can [import the backup file](#) into a test Jira instance to check the configuration.

### 1. Setting up the project

If you have a project in your target Jira instance that you wish to restore data into, you will need to ensure that the project is empty, i.e.

- no issues perform a search to find all issues in a project
- no components read the [Component management](#) page to find out how to view a summary of a project's components
- no versions read the [Version Management](#) page to find out how to view a summary of a project's versions

### 2. Setting up users and groups

The following types of users are considered mandatory for a project to be imported:

- reporter, assignee, component lead or project lead.

The following users are considered to be optional for a project to be imported:

- comment author/editor, work log author/editor, a user in a custom field (user picker), voter, watcher, change group author (i.e. someone who has changed an issue), attachment author, user in a project role.

The Project Import will attempt to create missing users if they are associated with the project. However, if the Project Import tool cannot create missing mandatory users in your target Jira instance, then you will not be permitted to import the project. This may occur if you have External User Management enabled in your target Jira instance you will need to disable External User Management or create the missing users manually in your external user repository before commencing the import.

 Please note that if you do not have enough information about the users in your backup file, the Project Import wizard will provide a link to a table of the missing users on a new page as well as a link to an XML file containing the missing users (on the new page). The table of users will display a maximum of 100 users, but the XML file will always be available.

### 3. Setting up custom fields

As described previously, the versions of your custom field apps must match between your backup and your target instance of Jira for your project to be imported. You need to ensure that you have set up your custom fields correctly in your target Jira instance, as follows:

- **Custom Field Type** If you do not have a particular [custom field type](#) (e.g. cascading select) installed on your target Jira, then all custom field data in your backup project that uses that custom field type will not be restored. However, your project can still be restored. For example, say you have a custom field, 'Title', which is a 'Cascading Select' field type and was

used in your backup project (i.e. there is saved data for this field). If you do not have the 'Cascading Select' custom field type installed on your target Jira, then all data for custom field 'Title' (and all other cascading select custom fields) will not be restored.

- **Custom Field Configuration** If you do have a particular [custom field type](#) (e.g. multi select) installed on your target Jira, then you must configure all of the custom fields (of that custom type) in your target Jira to match the equivalent custom fields in your backup project. Additionally, if your custom field has selectable options, then any options used (i.e. there is saved data for these options) in your backup project must exist as options for the custom field in your target Jira.

For example, say you have a custom multi select field named, 'Preferred Contact Method', in your backup project with options, 'Phone', 'Email', 'Fax'. Only the 'Phone' and 'Email' were actually used in your backup project. In this scenario, you need to set up your target Jira instance as follows:

- There must be a field named, 'Preferred Contact Method', in your target Jira instance.
- 'Preferred Contact Method' must be a multi select custom field type.
- 'Preferred Contact Method' must have the options, 'Phone' and 'Email' at a minimum, since they were used in your backup project. Please note, 'Preferred Contact Method' in your target Jira could also have additional options like 'Fax', 'Post', 'Mobile', etc, if you choose.

If you have not configured your existing custom field correctly, you will not be permitted to import your backup project until you correct the configuration errors in your target Jira.

See [Adding a custom field](#) for more information on custom field types and custom field configuration.

- **Compatibility with the Project Import tool** Custom fields also need to be compatible with the Project Import tool for the custom field data to be imported. Custom fields created prior to Jira v4.0 cannot be imported by the Project Import tool. The custom field developer will need to make additional code changes to allow the Project Import tool to restore the custom field data. If any of the custom fields used in your backup file are not compatible with the Project Import tool, the Project Import wizard will warn you and the related custom field data will not be imported. All the target Jira system custom fields and the custom fields included in Jira apps supported by Atlassian (e.g. Jira Toolkit, Charting app, Labels app, Perforce app) are compatible with the Project Import tool.

#### 4. Setting up workflows, system fields, groups and roles

In addition to custom fields, you need to correctly configure the project workflow, issue attributes (e.g. issue types) and groups/roles in your target Jira instance for your project to be restored successfully. Please ensure that you have reviewed the constraints on each of the following:

##### Workflows and workflow schemes:

- The project import process does not import workflows or workflow schemes. If you wish to retain a customized workflow from your backup, you will need to create a new workflow in your target Jira instance and manually edit the new workflow (e.g. create steps and transitions) to reflect your old workflow (note, the default Jira workflow is not editable). You will then have to add this workflow to a workflow scheme to activate it.
- When importing a project, the **Create Issue** transition and the **Issue Created** event will be triggered for each issue along with all corresponding [postfunctions](#). If you change the **Create Issue** transition after the import (for example, setting one of the create issue field values), you may get values that are different from those in the source. As a workaround, you can manually disable postfunctions during the import.
- Read more about creating and editing workflows in the [Working with workflows](#) and [Managing your workflows](#) documents. Please note that you may be required to create and edit a new workflow and workflow scheme to satisfy constraints on workflow entities from your backup, as described in the sections below, even if you do not wish to recreate the exact same workflow.

**i** Do not use the Jira functionality for exporting and importing workflow XML definitions, to copy your backup workflow to your target Jira instance. The workflow import/export tools do not include workflow screens in the process. Hence, you will be required to manually edit the workflow definitions post-import to match up new screens to the workflow, which is more work than it is worth.

##### Issue Types:

- If an [issue type](#) has been used in your backup project (i.e. there are issues of this issue type), you must set up the same issue type in your target Jira project. You may want to consider [setting up issue types for the project](#) instead of globally.

- **Workflow schemes** If you have associated an issue type with a particular workflow scheme in your backup project, you must ensure that the same association exists in your target Jira. See the above section on '**Workflow and Workflow Schemes**' for further information on how to set up a workflow in your target Jira instance.
- **Custom field configuration schemes** custom field configuration schemes can be used to apply a custom field configuration to specific issue types. If you have configured a custom field differently for different issue types in your backup project, you may wish to set up a custom field configuration scheme to apply the same custom field configuration to the same issue types in your target Jira instance. This will help ensure that you do not have a custom field for an issue type that is configured incorrectly (e.g. missing an option, if it has multiple selectable options), as described in the 'Setting up custom fields' section above.

#### Statuses:

- If an **issue status** has been used in your backup project (i.e. there are issues with the status), you must set up the same status in your target Jira project.
- **Workflow schemes** If you have linked a status into a particular workflow scheme in your backup project, you must ensure that the same association exists in your target Jira. See the above section on 'Workflow and Workflow Schemes' for further information on how to set up a workflow in your target Jira instance.

 Make sure to match the **Linked Status** name, not the **Step Name**, when inspecting your workflow.

#### Security Levels:

- If an **issue security level** has been used in your backup project (i.e. there are issues with this security level), it must be set up in your target instance of Jira. If you did not create an existing empty project, we recommend that you do so and set up the appropriate security levels for the project (via an issue security scheme).
- Issue security schemes Not applicable. It does not matter which users, groups or project roles are assigned to which security levels, as long as the appropriate security levels exist (please see the constraints on security levels in the 'Setting up entities and types' section).

#### Priority:

- If an **issue priority** has been used in your backup project (i.e. there are issues with this priority), it must be set up in your target instance of Jira.

#### Resolution:

- If an **issue resolution** has been used in your backup project (i.e. there are issues with this resolution), it must be set up in your target instance of Jira.

#### Issue Link Type:

- If an **issue link type** has been used in your backup project (i.e. there are issues associated by this link type), it must be set up in your target instance of Jira.

#### Project Role:

- If a **project role** has been used in your backup project (i.e. there are users/groups assigned to this project role), it must be set up in your target instance of Jira.  
(Note: The Project Import tool will copy across the **project role membership** from your backup project to your target Jira instance, if you choose. See the Project Import section for further details).

#### Group:

- If a **user group** has been used in your backup project (i.e. there are users in this group), it must be set up in your target instance of Jira.

 **A note about schemes**  
The project import process does not directly affect schemes, although entities and types associated

with schemes may be affected as described above. Please note that the following schemes are not affected at all by the project import:

- [Permission schemes](#) Not applicable. Permissions schemes do not need to match between the backup and target instance of Jira.
- [Notification schemes](#) Not applicable. Notification schemes do not need to match between the backup and target instance of Jira.
- [Screen schemes](#) Not applicable. Screen schemes do not need to match between the backup and target instance of Jira.
- [Issue type screen schemes](#) Not applicable. Issue type screen schemes do not need to match between the backup and target instance of Jira.
- [Field configuration schemes](#) Not applicable. Please note that if a field was configured as optional in your backup project and is configured as a required field in your target Jira instance, then the project will still be imported even if the field is empty. However, this field will be enforced as mandatory the next time a user edits an issue containing the field.

## 5. Setting up links

While the Project Import tool preserves the existing issue keys from your backed up project during the import process, the tool will also automatically create all issue links between issues within your backed up project. It will also try to create links between the backup project and another project, as long as the other project already exists in your target Jira instance with the relevant issue keys. If the source/target of a link cannot be found (i.e. the entire project or the particular issue may be missing), the link will not be created although the project will still be imported.

Note that the Project Import tool will create issue links between projects in either direction (source to target, or target to source). This means that if you import two projects from the same backup file, the second project import will create all of the links between the two projects that were missing from the first project import.

Once you have completed as many of the setup tasks as you are able to, run the [Project Import tool](#).

## Project Import

Restoring your project is a four step process:

1. [Specify the backup file](#)
2. [Select a project](#)
3. [Review data mapping validations](#)
4. [Verify the restored project](#)

If you start the Project Import tool, we strongly recommend that you complete all steps of the wizard before performing any other activities in Jira. Please be aware that it can take some time to validate the data mappings and then import the project.

You will most likely need to navigate away from the Project Import wizard to correct your Jira configuration, as advised by validation errors in the wizard. If you have to navigate to other pages in Jira to correct your Jira configuration or for other activities, you should:

- **(recommended)** open a separate session of Jira in a new browser window/tab. When you return to the Project Import wizard in the original browser window/tab, you can use the **'Refresh validations'** button on the validation screen to re-validate the data mappings; or,
- wait until the progress bar completes for the step you are currently in, before navigating elsewhere in Jira. The state of the Project Import wizard will be saved until you log out of Jira, your user session expires or you commence a different project import. You can resume your project import by returning to the Project Import page (via the main Administration menu) and selecting the 'resume' link on the first page of the wizard.

### 1. Specify the backup file

**Project Import: Select Backup File** ?

**i** This tool allows you to import a single JIRA project from a backup file. Importing a project into JIRA is a complex operation. It requires that you carry out manual modifications to the configuration of your JIRA instance. These modifications require that you have good understanding of, and experience in, JIRA administration and configuration.

**!** It is critical that you read our [Project Import documentation](#) and plan how to carry out the project import based on the information in that document. We strongly recommend that you first carry out the project import on a test JIRA instance, and then only carry it out on your production instance once you are sure that the test import was successful.

Please note that the backup file containing the project that you want to import must be from exactly the same version (6.0.3) of JIRA as this one.

While configuring a project import, JIRA will remain available to all users. Please be aware that once the data is actually being imported JIRA will be unavailable until the import has completed.

Please [backup](#) this JIRA instance before you begin the project import. The backup file and attachment paths must be located on the same machine as your JIRA instance.

File name\*  ?

Enter filename to restore project data from. Files will be loaded from : /home/rpillai/atlassian/application-data/jiralive/import

Backup Attachment Path **/home/jirouser/atlassian/application-data/jiralive/import/attachments**

Path Path to the directory holding backed up attachments.

**To start the Project Import tool:**

1. In the upper-right corner of the screen, select **Administration**  **> System**.
  2. Select **Import & Export > Project Import** to open the Project Import wizard page.
  3. Specify the path and name of your backup file in the 'File name' field. Your backup file must be an XML or ZIP file (as exported by Jira).
  4. Copy the attachments from the path where you have backed up the attachments to the **'Backup Attachment Path'** shown in the import window. This path is under the Jira home directory of the instance. Please note that if file attachments are not enabled in your target Jira instance you will not see the path to which you need to copy the attachments from the backup.
- Note:** You can choose to not copy the attachments to the **'Backup Attachment Path'**. If so, you will be able to restore your project from backup, however it will have no attachments associated with it. Please note, you cannot restore your attachments separately if you do not restore them as part of the project import, as the database entries for the attachments will be missing.

**2. Select a project to restore**

**Project Import: Select Project to Import** ?

The list of projects contains all projects present in the XML backup provided. Select the project you wish to import. The importer will attempt to automatically map the backup project's values to correct values in this instance of JIRA. Please make certain you have correctly configured the JIRA project in this instance (i.e. associated the correct schemes with the project, created any missing issue types, custom fields, etc.) See the documentation for full details of what needs to be done.

If a backup project can not be imported the details will be displayed below.

Projects from Backup

**!** No project with key 'DEMO' exists in this instance of JIRA. The importer will create a project with this key and the details of the backup project using the default schemes.

<b>Project:</b>	Demonstration Project
<b>Key:</b>	DEMO
<b>Description:</b>	<h3>Welcome to the administration of your demonstration project!</h3> <p>This is where you can view and change how the project is configured. Use the tabs on the left to navigate to different project settings.</p>
<b>Lead:</b>	admin
<b>URL:</b>	
<b>Sender Address:</b>	
<b>Default Assignee:</b>	Project Lead
<b>Issues:</b>	6
<b>Components:</b>	0
<b>Versions:</b>	0

1. Select a project to restore from the **'Projects from Backup'** drop-down. This drop-down will list all of the projects contained in your backup file.
2. If you have a valid project to restore from your backup, and your target Jira instance has an existing empty project, then the **'Overwrite Project Details'** option will display. Select the **'Overwrite Project Details'** option if you want to overwrite the project details of the existing empty project with the project details from your backup. The project details are the Name, URL, Project Lead, Default Assignee and Description of the project, as well as any [project role members](#) set up on your project. If there is no existing empty project in your target instance of Jira, this option will be checked and disabled as the Project Import will create the project with project details from your backup file.

### 3. Review data mapping validations

**Project Import: Pre-Import Summary - Demonstration Project** ?

The results of automatic mapping are displayed below. You will not be able to continue if any validation errors were raised.

- [Refresh validations](#) - re-maps and validates the backup data against the current state of JIRA.

**Errors**

- The data mappings have produced errors, you can not import this project until all errors have been resolved. See below for details.

<p><b>System Fields</b></p> <ul style="list-style-type: none"> <li><span style="color: green;">✔</span> Issue Type</li> <li><span style="color: red;">✘</span> Custom Field Configuration           <ul style="list-style-type: none"> <li>The custom field 'Ashamedly Khoikhoi's' of type 'Date Time' is required for the import but does not exist in the current JIRA instance.</li> <li>The custom field 'Calisthenics's' of type 'Number Field' is required for the import but does not exist in the current JIRA instance.</li> <li>The custom field 'Demeter's adverbs' of type 'Free Text Field (unlimited text)' is required for the import but does not exist in the current JIRA instance.</li> <li>The custom field 'Heartburn firepower Okhotsk's' of type 'Multi Select' is required for the import but does not exist in the current JIRA instance.</li> <li>The custom field 'Pedalled' of type 'Free Text Field (unlimited text)' is required for the import but does not exist in the current JIRA instance.</li> </ul> </li> </ul> <table border="0" style="width: 100%;"> <tr><td>• Status</td><td>Not checked yet</td></tr> <tr><td>• Priority</td><td>Not checked yet</td></tr> <tr><td>• Resolution</td><td>Not checked yet</td></tr> <tr><td>• Users</td><td>Not checked yet</td></tr> <tr><td>• Project Role</td><td>Not checked yet</td></tr> <tr><td>• Project Role Membership</td><td>Not checked yet</td></tr> <tr><td>• Group</td><td>Not checked yet</td></tr> <tr><td>• Issue Link Type</td><td>Not checked yet</td></tr> <tr><td>• Issue Security Level</td><td>Not checked yet</td></tr> <tr><td>• Attachments</td><td>Not checked yet</td></tr> </table>	• Status	Not checked yet	• Priority	Not checked yet	• Resolution	Not checked yet	• Users	Not checked yet	• Project Role	Not checked yet	• Project Role Membership	Not checked yet	• Group	Not checked yet	• Issue Link Type	Not checked yet	• Issue Security Level	Not checked yet	• Attachments	Not checked yet	<p><b>Custom Fields</b></p> <table border="0" style="width: 100%;"> <tr><td>• Demeter's adverbs</td><td>Not checked yet</td></tr> <tr><td>• Pedalled</td><td>Not checked yet</td></tr> <tr><td>• Ashamedly Khoikhoi's</td><td>Not checked yet</td></tr> <tr><td>• Laciest</td><td>Not checked yet</td></tr> <tr><td>• Heartburn firepower Okhotsk's</td><td>Not checked yet</td></tr> <tr><td>• Calisthenics's</td><td>Not checked yet</td></tr> <tr><td>• Centralized</td><td>Not checked yet</td></tr> <tr><td>• Eugenie</td><td>Not checked yet</td></tr> </table>	• Demeter's adverbs	Not checked yet	• Pedalled	Not checked yet	• Ashamedly Khoikhoi's	Not checked yet	• Laciest	Not checked yet	• Heartburn firepower Okhotsk's	Not checked yet	• Calisthenics's	Not checked yet	• Centralized	Not checked yet	• Eugenie	Not checked yet
• Status	Not checked yet																																				
• Priority	Not checked yet																																				
• Resolution	Not checked yet																																				
• Users	Not checked yet																																				
• Project Role	Not checked yet																																				
• Project Role Membership	Not checked yet																																				
• Group	Not checked yet																																				
• Issue Link Type	Not checked yet																																				
• Issue Security Level	Not checked yet																																				
• Attachments	Not checked yet																																				
• Demeter's adverbs	Not checked yet																																				
• Pedalled	Not checked yet																																				
• Ashamedly Khoikhoi's	Not checked yet																																				
• Laciest	Not checked yet																																				
• Heartburn firepower Okhotsk's	Not checked yet																																				
• Calisthenics's	Not checked yet																																				
• Centralized	Not checked yet																																				
• Eugenie	Not checked yet																																				

Previous
Refresh Validations
Cancel

1. The Project Import wizard will attempt to validate the data mappings required to import your project from the backup file. You can review the validations at this step of the wizard and modify your target Jira instance as required.
  - A tick symbol ( ✔ ) means that there are no problems with mapping these entities.
  - An exclamation mark symbol ( ⚠ ) means that there are problems with the data mapping that you should review before importing the project, but the project can still be imported. For example, a missing optional user that cannot be created automatically by the Project Import tool.
  - A cross symbol ( ✘ ) means that there are problems with the data mapping that must be fixed before you can import the project. For example, an Issue Type that is used in the backed up project is missing in your target Jira instance.
2. The ['Preparing your target Jira instance'](#) section on this page lists the common data mapping errors.
3. Once you have resolved the data validation errors as required, click **'Import'** to commence the import of data from your backup file.

i The Project Import tool will lock out your instance of Jira during the actual data import (not during the validations), so please ensure that your instance does not need to be accessible during this time.

### 4. Verify the restored project

**Project Import: Results** ?

The project import completed successfully in 0 minutes.

<p><b>Project Summary</b></p> <p>Key: DEMO</p> <p>Description: &lt;h3&gt;Welcome to the administration of your demonstration project!&lt;/h3&gt; &lt;p&gt;This is where you can view and change how the project is configured. Use the tabs on the left to navigate to different project settings.&lt;/p&gt;</p> <p>Lead: Admin</p> <p>URL:</p> <p>Sender Address:</p> <p>Default Assignee: Project Lead</p> <p>Components: 0</p> <p>Versions: 0</p>	<p><b>Users</b></p> <p>No users were added during the import.</p> <hr/> <p><b>Project Roles</b></p> <p>Administrators: 0 users, 1 groups</p> <p>Developers: 0 users, 1 groups</p> <p>Users: 0 users, 1 groups</p> <hr/> <p><b>Issues</b></p> <p>Issues created: 6 out of 6</p> <p>No attachments were added during the import.</p>
--	--

OK

1. Once the Project Tool has finished running, click **'OK'** to navigate to the restored project. You should verify that the issues, components and versions have been restored correctly. You should also check that any custom field data and links have been restored correctly.
2. Check that your attachments were correctly restored from your attachments backup directory.

 The Project Import tool will add an entry to every imported issue's Change History, showing when the issue was imported. Note that old entries in the Change History, from before the import, are retained for historical purposes only. Old entries may contain inconsistent data, since the configuration of the old and new Jira systems may be different.

### What if something went wrong?

- If your project import **did not complete**, you can refer to the Jira log file. The Project Import tool will log details of the operation to this file, including any unexpected errors and exceptions, e.g. database locked out, disk full, etc.
- If your project import completed but **did not restore your project as expected**, you may wish to attempt to fix the problem manually in your target Jira instance. You may also wish to try deleting the project in your target Jira instance and re-importing it from backup, paying special note to any warning validations (e.g. users that will not be added automatically).

If you cannot resolve the problem yourself, you can contact us for assistance. Please see the **'Need help'** section below for details.

### Need help?

Need further help? You can raise a support request in the Jira project at <https://support.atlassian.com> for assistance from our support team. Please attach to the support case:

- the backup file you are trying to import projects from, and
- the following information from your target Jira instance:
  - your log file
  - an **XML backup** of your target Jira instance
  - a copy and paste of the **entire contents** of the **System Info** page (accessed via the **Administration** tab), so that we know the details of your Jira configuration.

You can [anonymize the XML backups](#) if your data contains sensitive information.

# Anonymising Jira application data

**i** Support requests are often resolved **significantly** faster if a data export is provided as it will allow our legendary supporters direct access to a copy of your instance. We understand that sometimes this may be a difficult option due to the sensitivity of your data and have written an anonymizing tool to handle this particular scenario.

## Anonymizing Jira Data

The Jira inbuilt backup functionality will produce a ZIP file containing either 1 or 2 XML files, depending on the version that is being used. These files are a copy of the entire contents of Jira's database, encoded in XML, that can be used to restore an instance - we have further detail on this in our [Automating Jira application backups](#) documentation.

As of Jira 4.4, the backup functionality will produce a ZIP file that contains 2 XML files. These files will be `activeobjects.xml` and `entities.xml`. Only `entities.xml` will need to be anonymized - please do not attempt to anonymize the `activeobjects.xml`. For versions prior to 4.4, only one XML file will be produced with the same naming convention as the ZIP it is compressed as (for example `1970-Jan-010001.zip` will expand to `1970-Jan-01--0001.xml`).

1. Ensure that the `JAVA_HOME` variable has been configured, as in our [Setting JAVA\\_HOME](#) documentation.
2. Download the [Jira Anonymizer](#).
3. Create a temporary directory.
4. Unzip the anonymizer in the temporary directory.
5. Unzip the Jira backup ZIP file (for example `1970-Jan-01--0001.zip`) in the temporary directory.
6. Anonymize the backup file with the below commands:

```
$ java -DentityExpansionLimit=2147480000 -DtotalEntitySizeLimit=2147480000 -Djdk.xml.  
totalEntitySizeLimit=2147480000 -Xmx2g -jar joost.jar <JIRA BACKUP>.xml anon.stx > <NAME OF  
ANONYMISED BACKUP>.xml
```

For example, this would be anonymizing a Jira backup with the naming convention from Jira 4.4+:

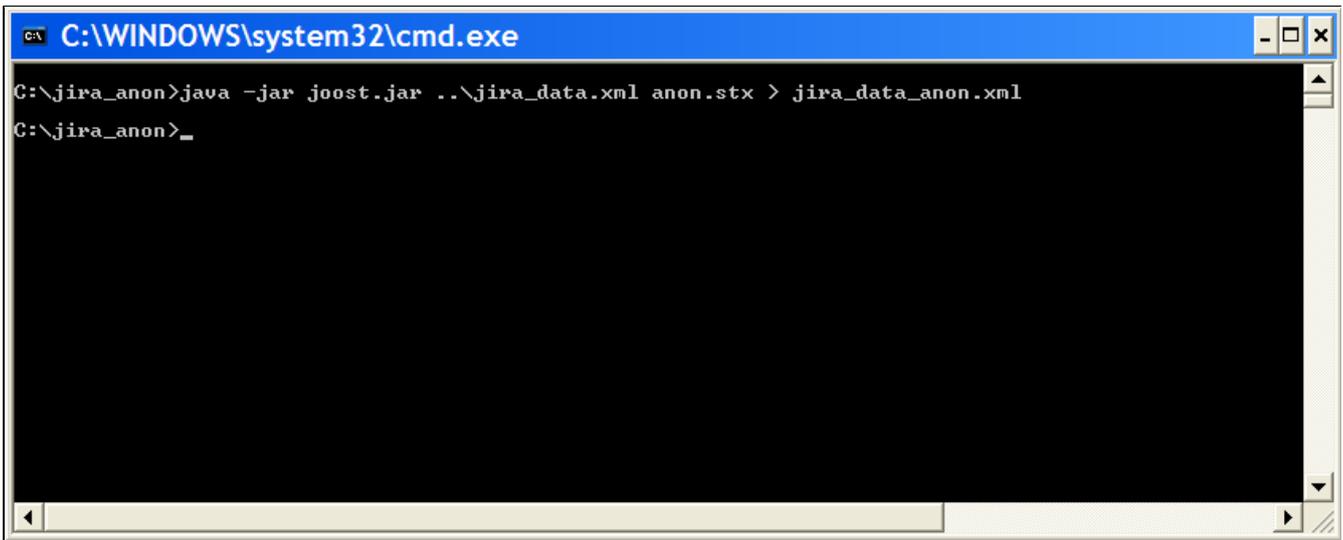
```
$ java -DentityExpansionLimit=2147480000 -DtotalEntitySizeLimit=2147480000 -Djdk.xml.  
totalEntitySizeLimit=2147480000 -Xmx2g -jar joost.jar entities.xml anon.stx > anon-entities.xml
```

**⚠** Depending on the size of the backup, additional memory may need to be allocated to the JVM. In order to do this, increase the value of the `Xmx` increments of 128m.

**i** If the Jira Anonymizer fails because of unsupported XML characters, you can clean up the `entities.xml` file with [Atlassian's XML cleanup utility](#). Once the file is cleaned up, run the Jira Anonymizer again.

7. Compress the generated anonymized XML backup file (e.g: `anon-entities.xml`) and the `activeobjects.xml` (*Jira 4.4.x + only*) into a ZIP or tarball.
8. Attach that ZIP or tarball onto the support issues as raised on [support.atlassian.com](#).
9. The temporary directory can now be removed.

The screenshot below is a simple example of how it is run in the command prompt of Windows XP:



```
C:\WINDOWS\system32\cmd.exe
C:\jira_anon>java -jar joost.jar ..\jira_data.xml anon.stx > jira_data_anon.xml
C:\jira_anon>_
```

## Information about the Anonymizer

The anonymizer currently replaces the following text with x's:

- Issue summary, environment, and description.
- Comments, work logs, change logs.
- Project descriptions.
- Descriptions for most elements (notification schemes, permission schemes, resolutions).
- Attachment file names.
- "Unlimited text" custom fields.

Please check the anonymized backup, `anon-backup.xml`, to ensure it's clean enough for the needs of your organization before sending it to Atlassian.

# Restoring data from an xml backup

## Before you begin

Make sure that you have the password to a login in the backup file that has the Jira System Administrator [global permission](#). Once the restoring procedure begins, all the existing data in the Jira application database is deleted, including all user accounts.

If you are restoring data from a Jira Cloud application site to a Jira Server application, please read [Migrating from Jira Cloud to Jira Server applications](#).

For all of the following procedures, you must be logged in with Jira System Administrator's [global permission](#).

## 1. Disable email sending/receiving

If you are restoring production data into a test Jira instance for experimentation purposes, you have to disable all Jira application's email features before you begin:

- **Disable email notifications** if Jira is configured to send emails about changes to issues, and you want to make test modifications to the copy, you should start Jira with the `-Datlassian.mail.senddisabled=true` flag.
- **Disable POP/IMAP email polling** if Jira is configured to poll a mailbox (to create issues from mails), you will have to disable polling on your test installation by setting the `-Datlassian.mail.fetchdisabled=true` flag.

Exactly how to set these flags is dependent on your particular application server, but for Jira, this is done by setting the `DISABLE_NOTIFICATIONS` environment variable before starting Jira (note, use `startup.sh` instead of `startup.bat` if you are not using Windows):

```
set DISABLE_NOTIFICATIONS=" -Datlassian.mail.senddisabled=true -Datlassian.mail.fetchdisabled=true -  
Datlassian.mail.popdisabled=true"  
cd bin  
startup.bat
```

You could also try un-commenting the `DISABLE_NOTIFICATIONS=" -Datlassian.mail.senddisabled=true -Datlassian.mail.fetchdisabled=true -Datlassian.mail.popdisabled=true"` line from your `/bin/setenv.bat` file (`/bin/setenv.sh` if you are not using Windows) and then running `startup`.

## 2. Restore the XML data

If you've used native database tools to back up your data, the restore process will be tool-specific and The stage 2 and 3 of these instructions don't apply to you.

1. In the upper-right corner of the screen, select **Administration**  **> System**.

2. Select **Import & Export > Restore System** to open the Restore Jira applications data from Backup page.

**Restore JIRA data from Backup**

i The backup file and index paths must be located on the same machine as your JIRA instance.  
You will be logged out after the restore process. Make sure you know your login details in the data being restored.

⚠ This will wipe all existing JIRA content - make sure you **backup first!**

File name\*

Enter a filename to restore data from. Files will be loaded from `:/data/jirastudio/jira/home/import`

Index Path **By default JIRA will use the index path specified in the backup file. The following path will be used for backups without an index path: `/data/jirastudio/jira/home/caches/Indexes`**

License (if required)

Only enter a license if you want to override the license in the import file.

Outgoing Mail  Enable  Disable

3. In the **'File name'** field, type the file name of the zipped XML backup file generated by Jira.
- i Ensure that this backup file has been moved or copied to the location specified below this field.
4. The **Index Path** field indicates where Jira will restore the search index data from the zipped XML backup file. This location (which cannot be modified) matches the index path specified in the zipped XML backup file. If, however, this backup file does not specify an index path, Jira will restore the search index to the `caches/indexes` subdirectory of the [Jira application home directory](#).
- ⚠ **Please Note:**
- The contents of the index directory may be deleted by the restore process.
  - The index directory should *only* contain Jira index data.
5. Click the **'Restore'** button and wait while your Jira data is restored.
- i Once the data has been restored, Jira will inform you that you have been logged out. This happens because all Jira users which existed in Jira prior to Jira's data being restored will have been deleted and replaced by users stored in the Jira export file.

i It is recommended that you avoid passing through a proxy when performing an XML restore, especially if your Jira instance is very large. Using a proxy may cause timeout errors.

### 3. Restore the attachments

If you created a backup of the attachments directory, you will need to restore the backup into a directory where Jira can access it.

⚠ If you use a custom directory for storing your attachments, ensure that Jira has read and write permissions to this directory and its subdirectories.

The process of restoring the attachments backup depends on the way it was created. Usually you can use the same tool to restore the backup as the one that was used to create it (see [Backing up attachments](#)).

If you are restoring the attachments into a different location (i.e. a different directory path) from where they were previously located (e.g. this will be the case when moving servers), please follow the instructions provided in [Configuring file attachments](#) to change the location of the attachments directory so that Jira can find the restored attachments

# Restoring information from a native backup

Before you begin

Make sure that you have the password to a login in the backup file that has the Jira System Administrator [global permission](#). Once the restoring procedure begins, all the existing data in the Jira application database is deleted, including all user accounts.

If you are restoring data from a Jira Cloud application site to a Jira Server application, please read [Migrating from Jira Cloud to Jira Server applications](#).

For all of the following procedures, you must be logged in with Jira Administrators [global permission](#).

## 1. Disable email sending/receiving

If you are restoring production data into a test Jira instance for experimentation purposes, you have to disable all Jira application's email features before you begin:

- **Disable email notifications** if Jira is configured to send emails about changes to issues, and you want to make test modifications to the copy, you should start Jira with the `-Datlassian.mail.senddisabled=true` flag.
- **Disable POP/IMAP email polling** if Jira is configured to poll a mailbox (to create issues from mails), you will have to disable polling on your test installation by setting the `-Datlassian.mail.fetchdisabled=true` flag.

Exactly how to set these flags is dependent on your particular application server, but for Jira, this is done by setting the `DISABLE_NOTIFICATIONS` environment variable before starting Jira (note, use `startup.sh` instead of `startup.bat` if you are not using Windows):

```
set DISABLE_NOTIFICATIONS=" -Datlassian.mail.senddisabled=true -Datlassian.mail.fetchdisabled=true -Datlassian.mail.popdisabled=true"
cd bin
startup.bat
```

You could also try un-commenting the `DISABLE_NOTIFICATIONS=" -Datlassian.mail.senddisabled=true -Datlassian.mail.fetchdisabled=true -Datlassian.mail.popdisabled=true"` line from your `/bin/setenv.bat` file (`/bin/setenv.sh` if you are not using Windows) and then running `startup`.

Follow these steps to restore data from a native backup.

1. Stop Jira
2. Replace the Jira Home directory with the backed up files.
3. Re-apply any changes made in the Jira Install directory

### Note

This step is required only if something has changed since the backup, it shouldn't contain any actual data.

4. Restore the database using native database tools (again this depends on the specific database type)
5. Start Jira

# Search indexing

To provide fast searching, Jira creates an index of the text entered into issue fields. This index is stored on the file system, and updated whenever issue text is added or modified. It's sometimes necessary to regenerate the index manually, for instance when you add a new custom field, or if the index is lost or corrupted. For more info on when you should re-index, see [Re-indexing after major configuration changes](#).

## On this page:

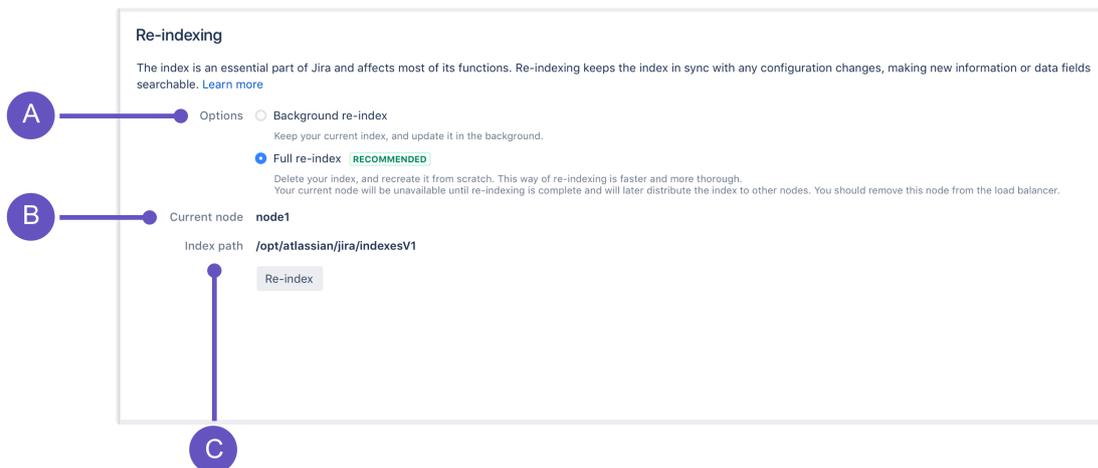
- [Re-indexing Jira](#)
- [Re-indexing options](#)
- [Choosing a custom Index Path](#)
- [Re-indexing Jira Data Center with no downtime](#)
- [Backing up and recovering your index](#)
- [Re-indexing a single project](#)

**i** For all of the following procedures, you must be logged in as a user with the **Jira Administrators** global permission.

## Re-indexing Jira

1. Go to **Administration** (⚙️) > **System**.
2. Select **Advanced** > **Indexing** to open the Indexing page.
3. This page allows you to choose one of the following two re-indexing options:

- **Background re-index:** Re-index all issues in the background.
- **Full re-index:** Delete and rebuild the whole index, including the comment and change history indexes.



- a. **Options:** *Background* or *full re-index*. If you're not sure which one to choose, see [Re-indexing options](#).
- b. **Current node:** Node on which the re-index will be performed. Available only in Jira Data Center.
- c. **Index path:** Directory where the index is stored.

## Re-indexing options

Not sure which re-indexing option to choose? Here's some information that will help you decide.

Background re-index	Full re-index
---------------------	---------------

Single-threaded, slower to complete.	Multi-threaded, faster to complete.
Can be canceled at any time.	Can't be canceled once started.
Keeps current index and updates it in-place.	Rebuilds the index, optimizes it, and deletes the old one.
Causes disk fragmentation.	Eliminates disk fragmentation.

### Background re-index

This option allows any Jira instance to remain usable during re-indexing; however, the instance will be slower. If you have to perform this option, do so during a low-usage period.

In addition, your disk fragmentation increases each time you perform a **Background re-index**. Whenever possible, perform a project re-index instead, especially if you just performed configuration changes that only affect one project. See [Re-indexing a single project](#) for instructions.

### Full re-index

Use this option when the indexes are corrupted, which may be caused by a system or disk failure. This option deletes all indexes and rebuilds them.

Overall, the **Full re-index** option provides greater benefits. The only downside for this option is that it'll lock a single-node instance, making it unavailable to users during the re-index.

 On a multi-node Jira Data Center, you can use the **Full re-index** option without actually locking the instance. Therefore, if your Data Center instance has multiple nodes, don't bother with a **Background re-index**. See [Re-indexing Jira Data Center with no downtime](#) for instructions.

Over time, your instance's disk becomes fragmented from normal use, which slows down your instance. Only the **Full re-index** option can address this. This means that, regardless of whether you're running a single- or multi-node instance, you should run a *full re-index* to address fragmentation periodically (for example, weekly or monthly).

 You can also speed up a full re-index by increasing the number of threads it uses. See [Tune number of index threads to make it go faster](#) for details.

Instances with lower data complexity also perform full re-indexes faster. See [Managing custom fields in Jira effectively](#) for related information.

## Choosing a custom Index Path

To change the Index Path, you must be logged in as a user with the **Jira System Administrators** [global permission](#).

- If you upgraded Jira with an [XML backup](#) from a Jira version prior to 4.2 and used a custom directory for your index path, you can choose between using this custom directory (which cannot be edited) or the default directory for your index path location. However, once you switch to using the default directory, you can no longer choose the custom directory option.
- The default directory location is the `caches/indexesV1` subdirectory of the [Jira application home directory](#).

 NFS storage for Jira indexes is not supported. See [Supported platforms](#) for more information.

### Re-indexing Jira Data Center with no downtime

Keeping the integrity of indexes is as important as having your Jira instance open to users all the time. These steps will help you run the **Full re-index** option, which deletes and recreates all indexes, with no downtime.

**Before you begin:**

Choose a node and remove it from the load balancer. You'll use it to perform the re-index.

**To re-index Jira Data Center with no downtime:**

1. Access Jira on the node you've chosen, and select **Administration**  **> System**.
2. Select **Advanced > Indexing** to open the Indexing page. Then, run **Full re-index**.
3. After the re-indexing is complete, take a look around the Jira instance to make sure everything looks fine.
4. Add the node back to the load balancer.

After completing the re-indexing, the rebuilt indexes will be automatically distributed to other nodes in the cluster (there might be some performance degradation during that time). If some changes were made to the indexes in the meantime, they will also be applied to maintain the integrity.

**Backing up and recovering your index**

Enabling index recovery will cause a snapshot of the indexes to be taken periodically. This allows you to recover your index quickly, rather than rebuilding the index, if there is a failure. This is particularly useful if you have a large Jira installation and you cannot afford for it to be offline for long. If you have a small Jira instance, it may not be worth enabling index recovery, as rebuilding the index won't take much time.

Whether a full index rebuild is faster than recovering from a snapshot depends on a number of factors, including how recent the snapshot being recovered was taken. Large and complex installations should test this process on a development/testing server before relying on it in production.

**To enable index recovery:**

1. Navigate to the **Indexing** page (as described above).
2. Click **Edit Settings** to enable index recovery and choose the frequency of snapshots.
  - Snapshots are stored in the `<yourJiraHome>/exports/export/indexsnapshotsdirectory`.

**To recover an index:**

1. Navigate to the **Indexing** page (as described above).
2. Enter the name of the previously saved index in **File name** and click **Recover**.
  - Jira will not be available during the recovery of the index.
  - If changes were made to the configuration that required a re-index after the snapshot was taken, then you will need to do a background re-index after the recovery. Note, Jira will be available after the recovery.

**Additional information**

- Jira will retain the last three snapshots at any time (in `<yourJiraHome>/exports/export/indexsnapshots`). Older snapshots will be automatically deleted. Note, snapshots may occupy considerable disk space and may need to be moved to offline storage or deleted as appropriate.
- The snapshot process is a relatively lightweight process and does not place much of a load on the system.
- The process of taking a snapshot will require temporary disk space equivalent to the index size. The resulting snapshots will each be about 25% the size of the index.
- All issues will be re-indexed appropriately during the recovery, including issues that were added, updated or deleted after the snapshot was taken.
- You can use the index recovery process to bring your index up to date, if you need to restore your Jira database. The index snapshot must pre-date the database backup being restored.

**Re-indexing a single project**

If you have made a configuration change that affects a single project, you can re-index just that project. See [Re-indexing after major configuration changes](#) for more information on when you should re-index.

If you have Jira System admin permissions, you can re-index a single project as follows:

1. In the upper-right corner of the screen, select **Administration**  > **Projects**.
2. Go to the project that you want to reindex.
3. Under the **Project settings** (the left-side panel), select **Re-index project**.
4. Select **Start project re-index**.

If you have Jira Project admin permissions, you can re-index a single project as follows:

1. In the bottom-right corner of the screen, select **Project settings** .
2. Under the **Project settings** (the left-side panel), select **Re-index project**.
3. Select **Start project re-index**.

 For more information about types of permissions that can be set up in Jira, go to [Permissions overview](#).

# Re-indexing after major configuration changes

Once issues have been created, modifying the configuration of your Jira instance can result in the search index becoming out-of-sync with Jira's configuration. Configuration details such as the following can affect the search index:

- [Field configuration schemes](#)
- [Custom fields](#)
- [Apps](#)
- [Time tracking](#)

If you make changes to any of these areas of configuration, you might see the following message in your Administration view:

```
USERFULNAME made configuration changes to 'SECTION' at TIME. It is recommended that you perform a re-index. It is recommended that you perform a re-index. For more information, please click the Help icon. To perform the re-index now, please go to the 'Indexing' section. Note: So that you only have to re-index once, you may wish to complete any other configuration changes before performing the re-index.
```

All users that have access to the Administration Tab will see this message (Jira Administrators, System Administrators, Project Administrators). The above message means that configuration changes have been made to Jira, but have not yet been reflected in the search index. Until Jira's search index has been rebuilt, it is possible that some search queries from Jira will return incorrect results. For example:

- If an app containing a custom field is enabled after being disabled, search queries which specify that the custom field should be empty will return *no issues* instead of *all issues*.
- If a [field configuration](#) is modified by altering the visibility of a particular field so that it is now visible, search queries which specify that field may also return erroneous results (depending on which field is being modified and what query is being executed).

The way to resolve the discrepancy is to [rebuild Jira's search index](#). This can take anywhere from seconds to hours, depending on the number of issues and comments in your Jira instance. While re-indexing is taking place, your instance will be unavailable to all users unless you chose Background Indexing. For these reasons, it is recommended that you:

- Make all your necessary configuration changes in one go before starting the re-index process; and
- Start the re-index process in a time period of low activity for your instance.

# Using robots.txt to hide from search engines

The [robots.txt protocol](#) is used to tell search engines (Google, MSN, etc) which parts of a website should not be crawled.

For Jira instances where non-logged-in users are able to view issues, a robots.txt file is useful for preventing unnecessary crawling of the Issue Navigator views (and unnecessary load on your Jira server).

## Editing robots.txt

Jira (version 3.7 and later) installs the following robots.txt file at the root of the Jira web app (`$JIRA-INSTALL/atlassian-jira`):

```
# robots.txt for Jira
# You may specify URLs in this file that will not be crawled by search engines (Google, MSN, etc)
#
# By default, all SearchRequestViews in the IssueNavigator (e.g.: Word, XML, RSS, etc) and all IssueViews
# (XML, Printable and Word) are excluded by the /sr/ and /si/ directives below.

User-agent: *
Disallow: /sr/
Disallow: /si/
```

Alternatively, if you already have a robots.txt file, simply edit it and add **Disallow: /sr/** and **Disallow: /si/**.

## Publishing robots.txt

The robots.txt file needs to be published at the root of your Jira internet domain, e.g. `jira.mycompany.com/robots.txt`.

**i** If your Jira instance is published at `jira.mycompany.com/jira`, change the contents of the file to `Disallow: /jira/sr/` and `Disallow: /jira/si/`. However, you still need to put robots.txt file in the root directory, i.e. `jira.mycompany.com/robots.txt` (not `jira.mycompany.com/jira/robots.txt`).

# Control anonymous user access

You can configure Jira to restrict anonymous user access and protect your data from being viewed by users that aren't logged-in. Oftentimes, anonymous users can access your data if you have filters or dashboards set to be viewed publicly or you have the Browse users global permission granted to the Anyone group. *Shared publicly* and *Anyone* in these cases mean anyone from **in and outside your organisation**.

Proper configuration can:

- prevent Jira filters, dashboards, project and user information from being shared unintentionally with the public by their owners (e.g. [Anonymous users able to see shared filters dashboards or project issues](#))
- control content that is available to search engine crawlers.

## Skip to:

- [I want to know if my instance has public facing content](#)
- [I want to restrict all my public facing content](#)

## I want to know if my instance has public facing content

1. Check whether public sharing is ON.

In order to identify if your instance has content that is open for the public you first need to confirm if the **Sharing with anyone on the web** functionality is ON. If your Jira instance is private and not to be used by users without logging in you should **turn this feature OFF**. This way users will no longer be able to share any dashboards or filters with anonymous users.

 This feature is OFF by default.

Options	
Allow users to vote on issues	ON
Allow users to watch issues	ON
Allow sharing filters/dashboards with anyone on the web	OFF
Maximum project name size	80
Maximum project key size	10
Allow unassigned issues	ON
External user management	OFF
Logout confirmation	Never
Use gzip compression	OFF
User email visibility	Public

Jira administrators can disable the option of Jira users to share dashboards and filters publicly. To disable this option do the following.

1. Go to **Administration** () > **System** > **General Configuration**.
2. Click **Edit Settings**.
3. Select OFF in **Sharing with anyone on the web**.
4. Click **Update**.

default.

Options

Voting  ON  OFF  
Allows users to vote on which issues they would like resolved.

Watching  ON  OFF  
Allows users to watch issues and keep notified of their progress.

Sharing with anyone on the web  ON  OFF  
Allows users to share dashboards and filters with anyone on the web (including users who are not logged in). Disabling this will not affect dashboards and filters that are already shared with anyone on the web. You'll need to update them manually. [Learn more](#)

Maximum project name size   
The minimum length is 2. 80 is recommended. Changing this length will not impact the currently existing project names.

 Turning the feature off will not affect existing filters and dashboards. If you change this setting, you will still need to update the existing filters and dashboards if they have already been shared publicly.

If you've still chosen to allow sharing with anyone on the web in some circumstances you can set the default preference for your users to *private* so that they can't accidentally grant public access to dashboards or filters.

To set the default sharing for filters and dashboards:

1. Go to **Administration**  > **System**.
2. Choose **Default user preferences** and select **Edit default values**.
3. Set **Default access** to *Private* and click **Update**.

2. Get the list of existing public filters and dashboards.

Even if you've disabled sharing with anyone on the web, this does not update existing filters and dashboards. To confirm if there are any filters or dashboards shared with the public we can run the SQL queries below.

 Always back up your data before performing any modifications to the database. If possible, test any alter, insert, update, or delete SQL commands on a staging server first.

## Filters

Get the list of all filters of the "A Share with anyone on the web" share type (i.e. global).

```
SELECT sr.filtername, sp.sharetype AS current_share_state, sr.username AS owner_name, sr.reqcontent
AS JQL
FROM searchrequest sr
INNER JOIN sharepermissions sp ON sp.entityid = sr.id
WHERE sp.sharetype='global' and sp.entitytype ='SearchRequest';
```

Alternatively go to **Administration**  > **System** > **Shared Items** > **Shared filters** and look for any filters with the **Shared with anyone on the web** status in the **Shared with** column.

## Dashboards

Get the list of dashboards of the "share with everyone" share type (i.e. global).

```
SELECT DISTINCT pp.id as Dashboard_Id, pp.pagename AS Dashboard_name, sp.sharetype AS
current_share_state, pp.username AS owner_name
FROM portalpage pp
INNER JOIN sharepermissions sp ON sp.entityid = pp.id
WHERE sp.sharetype='global' and sp.entitytype ='PortalPage'
ORDER BY pp.id;
```

Alternatively go to **Administration**  > **System** > **Shared Items** > **Shared dashboards** and look for any dashboards with the **Shared with anyone on the web** status in the **Shared with** column.

### 3. Monitor the Browse User global permission

Through the user picker functionality within Jira your user base information could be available to anonymous users. The Browse User [Global Permission](#) allows a user to view a list of all Jira user names and group names, share issues, and @mention people on issues. This is used for selecting users/groups in popup screens and also enables auto-completion of user names in most 'User Picker' menus and popups.

 If you grant this permission to the *Anyone on the web* group, you will be allowing anonymous users access to the endpoints that provide a list of users.

Ensure that this permission is restricted to specific groups that require it. You can restrict it in **Administration** (⚙️) > **System** > **Global Permissions**.

#### Global Permissions ?

These permissions apply to all projects. They are independent of project specific permissions. If you wish to set permissions on a project-by-project basis you can set them up in the [Permission Schemes](#). To allow users to log in, they must have [application access](#)

#### Jira Permissions ?

Permissions	Users / Groups
<b>Jira System Administrators</b> Ability to perform all administration functions. There must be at least one group with this permission.	<ul style="list-style-type: none"> <li>jira-administrators <a href="#">View Users</a> <a href="#">Delete</a></li> </ul>
<b>Jira Administrators</b> Ability to perform most administration functions (excluding Import & Export, SMTP Configuration, etc.).	<ul style="list-style-type: none"> <li>jira-administrators <a href="#">View Users</a> <a href="#">Delete</a></li> </ul>
<b>Browse Users</b> Ability to select a user or group from a popup window as well as the ability to use the 'share' issues feature. Users with this permission will also be able to see names of all users and groups in the system.	<ul style="list-style-type: none"> <li>jira-administrators <a href="#">View Users</a> <a href="#">Delete</a></li> <li>jira-servicedesk-users <a href="#">View Users</a> <a href="#">Delete</a></li> <li>jira-software-users <a href="#">View Users</a> <a href="#">Delete</a></li> </ul>

## I want to restrict all my public facing content

1. Update your existing public filters and dashboards

To restrict your public facing filters and dashboards, change the sharing configuration for the filters manually. To change filters in bulk, you need to perform an update through a DB query.

First, make sure you've turned off public sharing and identified content that is currently shared publicly.

2. Review the global **Browse Users** permission

Ensure that this permission is restricted to specific groups that require it.

- Go to **Administration** (⚙️) > **System** > **Global Permissions**.
- Go to the "**Browse Users**" permission under Jira Permissions.
- Remove the group "**Anyone on the web**" from this permission.

3. Review search engines crawlers access

See the document [Using robots.txt to hide from search engines](#) on how to control access from search engines.



After you restrict certain pages from Search Engine crawlers, the information that has already been cached by search engines may take a few days to become unavailable. There is no way to instantly remove all cached information from search engines besides contacting the service provider (e.g. Google) directly. You may need to provide evidence that you're the owner of the content that you're requesting to take down.

4. Block all anonymous access

You can use a dark feature to disable site-wide anonymous access was introduced. For more on dark features, see [Enable dark feature in Jira](#).

Enable the dark feature to disable public access

1. Login as an administrator and go to [BASE-URL]/secure/SiteDarkFeatures!default.jspa.
2. In the **Enable Dark Feature** text field add public.access.disabled.

# Licensing your Jira applications

You can view and manage your Jira application licenses on the **Versions & licenses** page. You may need to add or update your license if you:

- change the type of license (for example you may want to purchase a full license when your evaluation license expires)
- upgrade your user tier to accommodate new users
- add a new license when your old license has expired
- add a new license for a newly installed application

## Before you begin

 For all of the following procedures, you must be logged in as a user with the **Jira System Administrators** global permission.

## Updating your Jira license details

1. Obtain the license key you want to update (you can do this by visiting [my.atlassian.com](https://my.atlassian.com) or by contacting the member of your organization who handles IT product licensing).
2. Select **Administration**  **> Applications**.
3. Select **Versions & licenses** to view license details for your installed Jira applications.
4. Locate the license you want to update, and select the Edit icon  to edit the license.
5. Replace the existing license key with your new license key.
6. Select the **Update license** button to update the Jira application with the new license.

**JIRA Software 7.0.0** Unlimited users (15 used) 

<b>Trial expires</b>	28/Oct/15
<b>Support entitlement number (SEN)</b>	SEN-L1234567
<b>License type</b>	Evaluation
<b>Organisation name</b>	Example Company
<b>License key</b>	<input type="text"/>

**Update license** **Cancel**

 The updated license must be **compatible** with any other license(s) on your Jira server.

## Viewing your licensed user count

1. Select **Administration**  **> Applications**.
2. Select **Versions & licenses** to view license details for your installed Jira applications.
3. Next to the application name you can view the number of licensed users, as well as the number of users already used.

When you are approaching (or have exceeded) the maximum number of users for your license, a warning banner is displayed:



**⚠️ If you exceed the user count allowed by your Jira application's license, your users will not be able to create issues.** To prevent this, either upgrade to a larger license or reduce your existing user count.

## Upgrading to a larger license

### **i** Important changes to our server and Data Center products

We've ended sales for new Server licenses, and will end support for server on February 15, 2024. We're continuing our investment in Data Center with several key improvements. [Learn what this means for you](#)

1. Select **Administration**  **> Applications**.
2. Select **Versions & licenses** to view license details for your installed Jira applications.
3. Locate the license you want to upgrade.
  - a. If you're near the maximum users for the license, you'll see a banner prompting you to upgrade. Select the **Upgrade** button to be redirected to the Atlassian order form.
  - b. If you don't see the banner but still want to upgrade, visit the [Atlassian order form](#) directly.
4. Complete your purchase at the desired user tier. When you're finished, you'll have a new license key in your [my.atlassian.com](#) account.
5. Return to the **Versions & licenses** page and update your application with the newly purchased license key.

## Reducing your user count

You may want to reduce the user count for a Jira application if you have *exceeded your user count* or if you want to *change to a lower-tier license* to reduce costs.

The *recommended method* for reducing your user count in Jira is to remove users from the groups associated with the application. Remember a user may be a member of multiple groups, but will only count as one user on your license. See [Managing users](#) for more information.

Alternatively, if you have [connected Jira to an LDAP directory](#), you may want to configure Jira to synchronize a subset of users from LDAP rather than all users. See [Reducing the number of users synchronized from LDAP to Jira applications](#) for more information. However, this can be a complicated procedure and we recommend that you do not use this method unless necessary.

**⚠️ Note**, if you exceed the user count allowed by your Jira application's license, your users will not be able to create issues.

# Viewing your system information

Jira provides you with detailed information about your system configuration, as described in the table below. This information can be useful when modifying, troubleshooting, or upgrading your system.

## Viewing your Jira system information

1. Log in as a user with the '**Jira Administrators**' [global permission](#).
2. Choose **Administration**()>**System**. Select **System support > System Info** to open the System Info page.

The following categories of information is shown on the 'System Info' page:

### Warnings

Any warnings about known issues with your configuration will be displayed here.

### System info

Setting	Description
<b>Base URL</b>	The base URL of this Jira installation. It is used in outgoing <a href="#">email notifications</a> as the prefix for links to Jira issues. It can be changed as described in <a href="#">Configuring Jira options</a> .
<b>System Date</b>	The Jira server's system date.
<b>System Time</b>	The Jira server's system time.
<b>Current Working Directory</b>	States the current Jira Working Directory. Please see <a href="#">Important directories and files</a> for more information.
<b>Java Version</b>	The Jira server's Java version.
<b>Java Vendor</b>	The Jira server's Java vendor.
<b>JVM Version</b>	The Jira server's JVM version.
<b>JVM Vendor</b>	The Jira server's JVM vendor.
<b>JVM Implementation Version</b>	The Jira server's JVM implementation version.
<b>Java Runtime</b>	The Jira server's Java runtime environment.
<b>Java VM</b>	The Jira server's Java Virtual Machine.
<b>User Name</b>	The operating system login name which Jira runs under.
<b>User Timezone</b>	The Jira server's timezone.

<b>User Locale</b>	The Jira server's locale. Unless the default language is modified in Jira's <a href="#">general configuration</a> , the User Locale will dictate the default language.
<b>System Encoding</b>	The Jira server's system encoding.
<b>Operating System</b>	The Jira server's operating system.
<b>OS Architecture</b>	The Jira server's operating system architecture (e.g. i386).
<b>Application Server Container</b>	The application server in which your Jira instance is running.(See <a href="#">Supported platforms</a> for a list of supported application servers.)
<b>Database type</b>	The type of database to which your Jira instance is connected.(See <a href="#">Supported platforms</a> for a list of supported application servers.)
<b>Database JNDI address</b>	The JNDI address of the database to which your Jira instance is connected.(For more details, see <a href="#">Connecting Jira to a database.</a> )
<b>Database URL</b>	The URL of the database to which your Jira instance is connected.(For more details, see <a href="#">Connecting Jira to a database.</a> )
<b>Database version</b>	The version of the database to which your Jira instance is connected.(See <a href="#">Supported platforms</a> for a list of supported application servers.)
<b>Database driver</b>	The driver which your Jira instance is using to connect to its database.(For more details, see <a href="#">Connecting Jira to a database.</a> )
<b>External user management</b>	'ON' / 'OFF' indicates whether Jira's users are being managed externally or internally to Jira (e.g. via <a href="#">Crowd</a> ).
<b>Crowd integration</b>	'YES' / 'NO' indicates whether Atlassian's <a href="#">Crowd identity management system</a> has been integrated with this instance of Jira. For more information please see the chapter titled ' <a href="#">Integrating Jira with Crowd</a> ' in the Crowd documentation.
<b>JVM Input Arguments</b>	A list of any variables that are being passed to your application server when it starts up.(For more information, see <a href="#">Setting properties and options on startup.</a> )
<b>Modified Files</b>	A list of any files in your Jira installation that have been modified as part installation or customization of Jira.
<b>Removed Files</b>	A list of any files that have been removed from your Jira installation.

### Java VM Memory Statistics

Java applications, such as Jira, run in a "Java virtual machine" (JVM) instead of directly within an operating system. When started, the Java virtual machine is allocated a certain amount of memory, which it makes available to applications like Jira. The following table shows the JVM memory data for your Jira instance.

Setting	Description
<b>Total Memory</b>	The total amount of memory allocated to the JVM that is available to this instance of Jira.(For more details, see <a href="#">Increasing Jira memory.</a> )

<b>Free Memory</b>	The amount of free JVM memory currently available to this instance of Jira.
<b>Used Memory</b>	The amount of JVM memory currently being used by this instance of Jira.
<b>Total PermGen Memory</b>	The total amount of PermGen (Permanent Generation) memory available to this instance of Jira.
<b>Free PermGen Memory</b>	The amount of free PermGen (Permanent Generation) memory currently available to this instance of Jira.
<b>Used PermGen Memory</b>	The amount of PermGen (Permanent Generation) memory currently being used by this instance of Jira.
<b>Memory Graph</b>	A bar graph showing the available versus free JVM memory. You can click the ' <b>Force garbage collection</b> ' link to start a clean-up. Note that <b>this is generally not needed (even if the graph shows 100% utilization)</b> unless you want to examine Jira's baseline heap usage.
<b>PermGen Memory Graph</b>	A bar graph showing the available versus free PermGen (Permanent Generation) memory.
<b>Non-Heap Memory Graph (includes PermGen)</b>	A bar graph showing the available versus free non-heap memory (including PermGen memory).

You can click the '**More Information...**' link at the bottom of this table to view an additional section titled '**Memory Pool Info**' (which lists detailed information about the various parts of memory that the Java virtual machine uses to store its data, and is generally only useful to Atlassian's support engineers.)

### Jira Info

Setting	Description
<b>Uptime</b>	The period of time since your Jira instance was last started.
<b>Edition</b>	The 'edition' of Jira you are running.
<b>Version</b>	The version of Jira you are running.
<b>Build Number</b>	The build number of your Jira version. This is generally only useful to Atlassian's support engineers.
<b>Build Date</b>	The date on which your Jira version was built. This is generally only useful to Atlassian's support engineers.
<b>Atlassian Partner</b>	Indicates whether your distribution of Jira was built by an Atlassian partner company. Blank indicates that it was built directly by Atlassian.
<b>Installation Type</b>	Indicates whether Jira has been installed as a ' <b>recommended</b> ' distribution or as a 'WAR' distribution. (Note we no longer support WAR installations or builds.)
<b>Server ID</b>	This number is calculated automatically by Jira, based on your license number.

<b>Last Upgrade</b>	The time at which your Jira installation was last upgraded, and from which version it was upgraded from (if applicable). Click the ' <b>More Information...</b> ' link to see a list of all upgrades that have been performed on your JIRA system from version 4.1 onwards.
<b>Installed Languages</b>	A list of all language packs available within the Jira system.(Note: to install additional languages, see <a href="#">Translating Jira.</a> )
<b>Default Language</b>	The language used throughout the Jira interface. To change the default language, see <a href="#">Configuring Jira options</a> . Note that users can override the default language by changing the <b>Language</b> preference in their user profile.

## License Info

⚠ To edit your license details, see [Licensing your Jira applications](#). Note that you will require the '**Jira System Administrators**' [global permission](#).

Setting	Description
<b>Date Purchased</b>	The date on which this system's Jira license was originally purchased. Note: you can verify this information by visiting <a href="http://my.atlassian.com">http://my.atlassian.com</a>
<b>License Type</b>	For information about the different types of Jira licenses, please see <a href="http://www.atlassian.com/software/jira/licensing.jsp">http://www.atlassian.com/software/jira/licensing.jsp</a>
<b>Maintenance Period End Date</b>	For information about Jira support and maintenance, please see <a href="http://www.atlassian.com/software/jira/licensing.jsp">http://www.atlassian.com/software/jira/licensing.jsp</a>
<b>Maintenance Status</b>	For information about Jira support and maintenance, please see <a href="http://www.atlassian.com/software/jira/licensing.jsp">http://www.atlassian.com/software/jira/licensing.jsp</a>
<b>Support Entitlement Number (SEN)</b>	For information about Jira support and maintenance, please see <a href="http://www.atlassian.com/software/jira/licensing.jsp">http://www.atlassian.com/software/jira/licensing.jsp</a>

## Configuration Info

Setting	Description
<b>Attachments Enabled</b>	'true' / 'false' indicates whether or not users can attach files and screenshots to issues in this Jira system (subject to <a href="#">project permissions</a> ). For more information, see <a href="#">Configuring file attachments</a> .
<b>Issue Voting Enabled</b>	'true' / 'false' indicates whether or not users can vote on issues in this Jira system (subject to project permissions). For more information, see <a href="#">Configuring Jira options</a> .
<b>Issue Watching Enabled</b>	'true' / 'false' indicates whether or not users can watch issues in this Jira system (subject to project permissions). For more information, see <a href="#">Configuring Jira options</a> .
<b>Unassigned Issues Enabled</b>	'true' / 'false' indicates whether or not issues can be 'unassigned' (i.e. assigned to no one) in this Jira system. For more information, see <a href="#">Configuring Jira options</a> .

<b>Sub-Tasks Enabled</b>	'true' / 'false' indicates whether or not 'sub-task' issues can be created in this Jira system. For more information, see <a href="#">Configuring sub-tasks</a> .
<b>Issue Linking Enabled</b>	'true' / 'false' indicates whether or not issues can be linked to each other within this Jira system. For more information, see <a href="#">Configuring issue linking</a> .
<b>Time Tracking Enabled</b>	'true' / 'false' indicates whether or not time (work) can be logged on issues in this Jira system. For more information, see <a href="#">Configuring time tracking</a> .
<b>Time Tracking Hours Per Day</b>	The number of hours per working day for which work that can be logged on issues in this Jira system. For more information, see <a href="#">Configuring time tracking</a> .
<b>Time Tracking Days Per Week</b>	The number of days per week for which work that can be logged on issues in this Jira system. For more information, see <a href="#">Configuring time tracking</a> .

### Database statistics

The information in this section can help determine how much resource (e.g. memory) your Jira system requires.

Setting	Description
<b>Issues</b>	The number of issues that have been created in this Jira system.
<b>Projects</b>	The number of projects that have been created in this Jira system.
<b>Custom Fields</b>	The number of <a href="#">custom fields</a> that have been created in this Jira system.
<b>Workflows</b>	The number of <a href="#">workflows</a> that have been created in this Jira system.
<b>Users</b>	The number of user IDs that have been created in this Jira system.
<b>Groups</b>	The number of groups that have been created in this Jira system.

### File Paths

Setting	Description
<b>Location of Jira Home</b>	The path to your Jira home directory.(For information about changing the location, see <a href="#">Setting your Jira application home directory</a> .)
<b>Location of entityengine.xml</b>	The path to your Entity Engine.(For database-specific information about configuring your entityengine.xml file, see <a href="#">Connecting Jira applications to a database</a> .)
<b>Location of atlassian-jira.log</b>	The path to the Jira log file. Note that, if you are requesting support, the support engineers will generally need your application server log file as well as your Jira log file.(For information about changing the logging level, see <a href="#">Logging and profiling</a> ; note that you will require the <b>'Jira System Administrators'</b> <a href="#">global permission</a> .)

<b>Location of indexes</b>	The path to your Jira search indexes, not your database indexes. (For information about moving the indexes, please see <a href="#">Search indexing</a> ; note that you will require the <b>'Jira System Administrators'</b> <a href="#">global permission</a> .)
----------------------------	--

## Listeners

This section lists all the listeners that are installed in this Jira system. For more information, see [Listeners](#). Note that you will require the **'Jira System Administrators'** [global permission](#) in order to register a listener.

## Services

This section lists all the services that are installed in this Jira system. For more information, please see [Services](#). Note that you will require the **'Jira System Administrators'** [global permission](#) in order to register a service.

## Apps

The app sections lists all plugins that are installed in this Jira system, broken down by System Apps and User installed Apps. For more information, please see [Managing apps](#).

## System properties

The information in this section is specific to the application server and Java version you are using, and is generally only useful to Atlassian's support engineers.

## Trusted Applications

This section lists all 'trusted application' (i.e. applications that Jira will allow to access specified functions on behalf of any user without the user logging in to Jira). Trusted applications have now been superseded by application links, and you can find more information on [application links](#) here.

# Live monitoring using the JMX interface

This article describes how to expose JMX MBeans within Jira for monitoring with a JMX client.

 This guide provides a basic introduction to the JMX interface and is provided as is. Our support team can help you troubleshoot a specific Jira problem, but aren't able to help you set up your monitoring system or interpret the results.

## What is JMX?

JMX ([Java Management Extensions](#)) is a technology for monitoring and managing Java applications. JMX uses objects called MBeans (Managed Beans) to expose data and resources from your application. For large instances of Jira Server or Jira Data Center, enabling JMX allows you to more easily monitor the consumption of application resources and [diagnose performance issues related to indexing](#). This enables you to make better decisions about how to maintain and optimize machine resources.

JMX attribute name	Description
50thPercentile	The value at the 50th percentile in the distribution of measured times (the median value).
75thPercentile	The value at the 75th percentile in the distribution of measured times.
95thPercentile	The value at the 95th percentile in the distribution of measured times.
98thPercentile	The value at the 98th percentile in the distribution of measured times.
99thPercentile	The value at the 99th percentile in the distribution of measured times.
999thPercentile	The value at the 99.9th percentile in the distribution of measured times.
Count	The number of invocations since node startup.
DurationUnit	The time unit used to report percentile values, min, max, mean and standard deviation. Default is milliseconds.
FifteenMinuteRate	The fifteen-minute moving average rate of invocations since node startup. This rate uses the same exponential decay factor as is used for the fifteen minute load average in <code>Unixstopcommand</code> .
FiveMinuteRate	The five-minute moving average rate of invocations since node startup. This rate uses the same exponential decay factor as is used for the five minute load average in <code>Unixstopcommand</code> .
Max	The highest measured time since node startup.
Mean	The mean measured time since node startup.
MeanRate	The mean rate of invocations since node startup.
Min	The lowest measured time since node startup.
OneMinuteRate	The one-minute moving average rate of invocations since node startup. This rate uses the same exponential decay factor as is used for the one minute load average in <code>Unixstopcommand</code> .

RateUnit	The unit in which MeanRate, OneMinuteRate, FiveMinuteRate and FifteenMinuteRate are reported. Default is events / second.
StdDev	The standard deviation in the measured times since node startup.

## Metrics collected by Jira

The following table lists metrics (MBeans) that are collected by Jira. All of them are grouped in the `com.atlassian.jira` property.

Metric	Description	Reset after restarting Jira
dashboard.view.count	The number of times all dashboards were viewed by users.	Yes
entity.attachments.total	The number of attachments.	-
entity.components.total	The number of components.	-
entity.customfields.total	The number of custom fields.	-
entity.filters.total	The number of filters.	-
entity.groups.total	The number of user groups.	-
entity.issues.total	The number of issues.	-
entity.users.total	The number of users.	-
entity.versions.total	The number of versions created.	-
issue.assigned.count	The number of times issues were assigned or reassigned to users (counts each action).	Yes
issue.created.count	The number of issues that you created after starting your Jira instance.	Yes
issue.link.count	The number of issue links created after starting your Jira instance.	Yes

issue. search. count	The number of times you searched for issues.	Yes
issue. updated. count	The number of times you updated issues (each update after adding or changing some information).	Yes
issue. worklogge d.count	The number of times you logged work on issues.	Yes
jira.license	The types of licenses you have, the number of active users, and the maximum number of users available for each license type.	-
quicksearc h. concurrent .search	The number of concurrent searches that are being performed in real-time by using the quick search. You can use it to determine whether the limit set for concurrent searches is sufficient or should be increased.	Yes
web. requests	The number of requests (invocation.count), and the total response time (total.elapsed.time).	Yes

The following metrics have been added to Jira as of 8.1 and are exposed under `com.atlassian.jira/metrics`. All the metrics mentioned below will be reset after restarting Jira.

Metric path	Description
<b>comment</b>	Metrics related to comment operations
comment /Create	Comment being created
comment /Delete	Comment being deleted
comment /Update	Comment being updated
<b>indexing</b>	Metrics related to issue, comment, worklog and change indexing
indexin g /Create Change History Docum ent	Index documents being created for Change History entities. Note that many Change History Documents can be created for every issue.
indexin g /Create Comme ntDocu ment	Index documents being created for Comment entities.

indexing /CreateIssueDocument	Index documents being created for Issue entities
indexing /IssueAddedFieldIndexers	FieldIndexer modules enrich Issue documents as part of Index document creation. Plugins can register customFieldIndexer modules. These metrics provide insight into how much time is spent inFieldIndexer, and can be used to track down indexing performance issues caused by them. The metrics describe how much time was spent in allFieldIndexers combined per Issue document created.
indexing /LuceneAddDocument	How much time was spent adding a document to the Lucene index
indexing /LuceneDeleteDocument	How much time was spent deleting one or more document matching a term from the Lucene index
indexing /LuceneOptimize	Metrics about Lucene index optimization (triggered manually from Jira)
indexing /LuceneUpdateDocument	How much time was spent adding a created document to the Lucene index.
indexing /ReplicationLatency	Replication latency is the time between an issue, comment or worklog being indexing on the node where the change was made and the indexing operation being replayed on the current node.
indexing /WaitForLucene	Documents are written to the Lucene index asynchronously. This metric captures how much time Jiras indexing thread spent waiting for Lucene to complete the write.
indexing /issueAddedSearchExtractors	EntitySearchExtractor enrich issue documents as part of Index document creation. Plugins can register customEntitySearchExtractor modules. These metrics provide insight into how much time is spent inEntitySearchExtractor, and can be used to track down indexing performance issues they cause. The metrics describe how much time was spent in allEntitySearchExtractors combined per Issue document created.
<b>issue</b>	<b>Metrics about issue operations</b>
issue /Create	Issue being created
issue /Delete	Issue being deleted
issue /Index	Issue being added to the Lucene index. This covers issue document creation and adding the document to the index.

issue /DeIndex	Issue being removed from the Lucene index
issue /ReIndex	Issue being re-indexed (as a result of issue updates). This covers issue document creation, deleting the old document from the index and adding new documents to the index.
issue /Update	Issue being updated

## Monitoring Jira

Before you can monitor Jira, you need to enable JMX monitoring and then use a JMX client to view the metrics.

### Good to know

Viewing the metrics will always have some performance impact on Jira. We recommend that you don't refresh them more than once a second.

## Enabling JMX monitoring in Jira

All of the metrics are collected by default, but you need to enable JMX monitoring to expose them. You can do it in Jira, but you need to be a Jira administrator to get there.

1. In Jira, go to **Administration**  > **System** > **JMX monitoring**.
2. Toggle **Enable JMX monitoring**.

## Monitoring with JConsole

After you enabled JMX monitoring, you can use any JMX client to view the metrics. To make it quick and easy, we've described how to view them by using JConsole. You can monitor your Jira instance either locally, or remotely:

- **Monitoring Jira locally** is good if you're troubleshooting a particular issue, or only need to monitor Jira for a short time. Local monitoring can have a performance impact on your server, so it's not recommended for long-term monitoring of your production system.

To monitor locally:

1. Start JConsole (you'll find it in the `bin` directory of the JDK installation directory).
2. Select **Local Process**.
3. Select the Jira process. It will be called something like `org.apache.catalina.startup.Bootstrap start`.
4. After connecting, expand the `com.atlassian.jira` property that groups all the metrics.

See [Using JConsole](#) for more information on local monitoring.

- **Monitoring Jira remotely** is recommended for production systems, as it does not consume resources on your Jira server.

To monitor remotely:

1. Add the following properties to your `setenv.sh`/`setenv.bat` file. The port can be any port that is not in use.

#### Windows...

```
set CATALINA_OPTS=-Dcom.sun.management.jmxremote %CATALINA_OPTS%
set CATALINA_OPTS=-Dcom.sun.management.jmxremote.port=8099 %CATALINA_OPTS%
```

**Linux...**

```
CATALINA_OPTS="-Dcom.sun.management.jmxremote ${CATALINA_OPTS}"  
CATALINA_OPTS="-Dcom.sun.management.jmxremote.port=8099 ${CATALINA_OPTS}"  
export CATALINA_OPTS
```

2. Decide how you will secure your remote connection. See [Remote Monitoring and Management](#) for more information.  
Although it is possible to disable authentication, we do not recommend doing this on a production system.
3. Start JConsole (you'll find it in the `bin` directory of the JDK installation directory).
4. Select **Remote Process**.
5. Enter your hostname and port (this is the port you specified earlier, not the Jira port).
6. Click **Connect**.
7. After connecting, expand the `com.atlassian.jira` property that groups all the metrics.

See [Using JConsole](#) for more information on remote monitoring.

# Monitoring database connection usage

Jira provides a view of its database connection usage. This provides information on the activity of the connection pool, as well as the frequency of reads/writes to the database. You can use this information to tune your database connections for better performance.

The instructions on this page describe how to navigate to the database connection usage information in the Jira administration console, and how to interpret the information. If you want to make changes to your database connection pool settings using this information, see this related topic: [Tuning database connections](#).

**Note:**For all of the following procedures, you must be logged in as a user with the **Jira Administrators** [global permission](#).

## Data Center monitoring

If you're running Jira Data Center, see the [latest available monitors](#).

### On this page:

- [Accessing the Database Monitoring page](#)
- [Interpreting the database monitoring graphs](#)

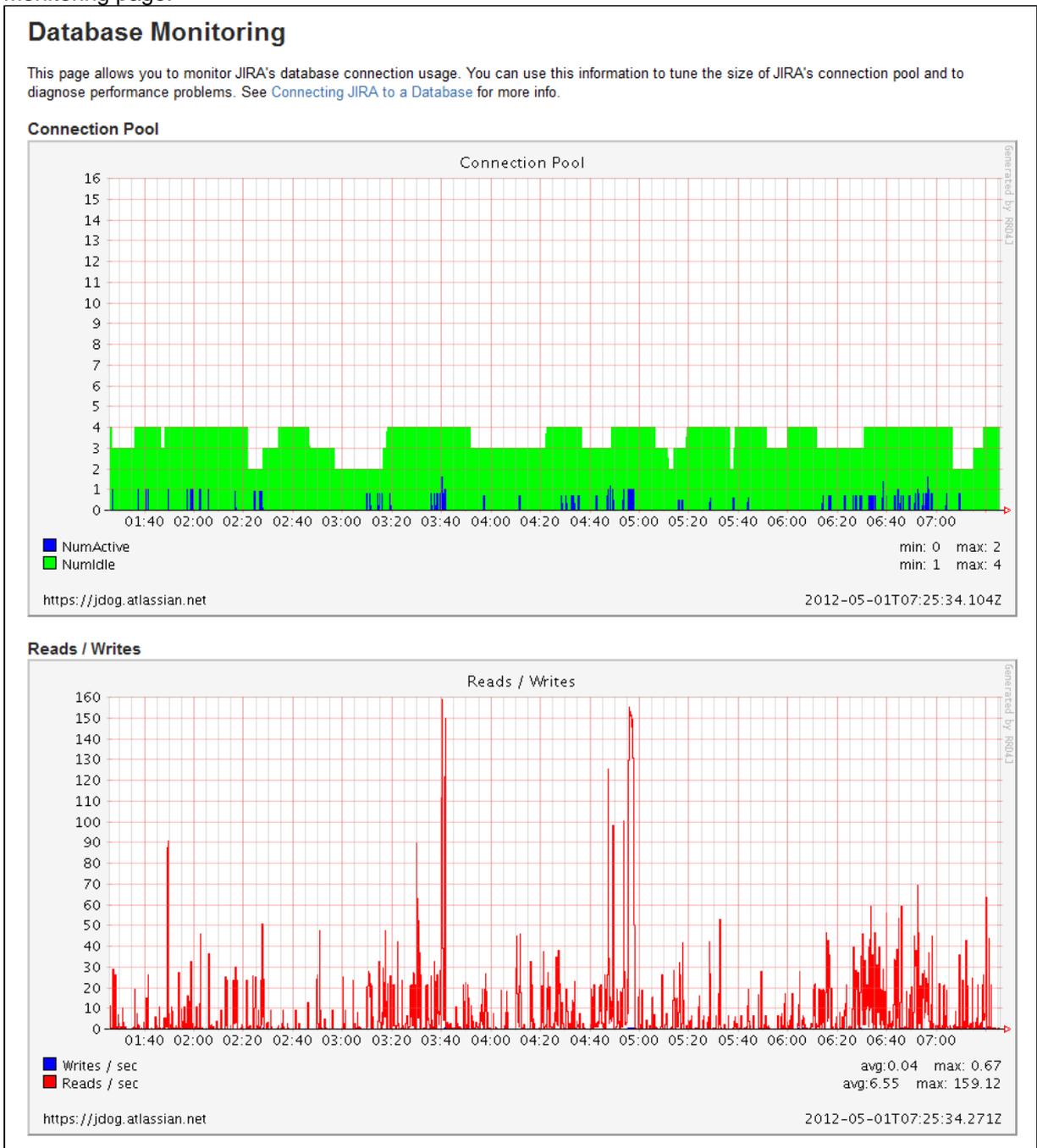
### Related pages:

- [Tuning database connections](#)
- [Enterprise Resources](#)

## Accessing the Database Monitoring page

1. Choose **Administration** () > **System**.

2. Select **Database Monitoring**, which can be found under **System support** to display the Database Monitoring page.



## Interpreting the database monitoring graphs

### Connection Pool graph

The 'Connection Pool' graph shows the activity in the connection pool for the last 6 hours.

- This graph shows the number of active and idle connections, as well as the maximum and minimum for the period.
- The scale of the vertical axis is equal to the maximum number of connections.
- The readings are averages over a period of 5 minutes.

This information can help you to optimize database connection usage. For example, if the number of active connections is consistently or frequently near to the maximum available, then you may need to raise the maximum connections available in the pool. Conversely, if the number of active connections is consistently low compared to the maximum available, then you may want to lower the maximum connections available in the pool. For more information on how to tune database connections, see [Tuning database connections](#).

**Reads / Writes graph**

The 'Reads / Writes' graph shows the frequency of reads and writes to the database over a period of time. It can be helpful to correlate database usage with connection pool usage. Whenever Jira needs to access (i.e. read from or write to) the database, a database connection is required. If there are regular spikes in the reads / writes, you may need to consider raising the maximum connections available in the pool.

# Monitor your instance with Jira diagnostics plugin

When investigating a performance problem or outage, it's useful to know as much as possible about what was happening in your site in the lead-up to the problem. This is when diagnostics information can help. That is why we produce a daily summary on all alerts fired with the Atlassian diagnostics framework.

While often not individually actionable, diagnostic alerts can help you build up a detailed picture of your application's behaviour, and identify symptoms that may be contributing to the problem.

 The plugin is available in Jira version 7.13 and later.

## About diagnostic alerts

The purpose of the diagnostics tool is to continuously check for symptoms or behaviours that we know may contribute to problems in your application. An alert is triggered when a set threshold is exceeded.

Property	Default value	Explanation
<code>jira.diagnostics.thresholds.slow-query-millis</code>	400 ms	Alert for slow JQL query
<code>jira.diagnostics.thresholds.number-of-results</code>	1000	Number of returned issues from a single JQL query
<code>jira.diagnostics.thresholds.query-complexity</code>	1000	Lucene query complexity - number of clauses the query is constructed of

For example, if a query takes more than 0.4 s, an alert is triggered. This is useful because if users run a lot of queries at the same time, it might overload the system.

It's important to note that the thresholds are just the point at which the alert is triggered. It's not the same as a timeout, or other hard limit. For example, a long running task may trigger an alert after 5 minutes, and still complete successfully after 8 minutes.

When an alert is triggered, a message is written to `theatlassian-jira.logfile` (your application log), and further details are provided in `thejira-diagnostics.logfile`. It's also included in support zips.

## Sample alerts

Log entry	Explanation
-----------	-------------

<pre>2020-06-04 03:27: 15,009+0000 pool-23- thread-1 INFO ServiceRunner [atlassian- diagnostics-data- logger] 1591241235004 ; INFO ; DB ; DB-3002 ; High database con nection pool utilization detected. ; &lt;app- key&gt; ; ; ; {"activeConnections" :2 1,"idleConnections": 4,"maxConnections": 25}</pre>	<p>The database connection pool was been highly utilized in the last sampling period.</p>
<pre>INFO ServiceRunner [atlassian- diagnostics-data- logger] 1591241235004 ; INFO ; DB ; DB-3002 ; Slow scheduled job ;</pre>	<p>A scheduled job took longer than its configured interval to complete.</p>
<pre>INFO ServiceRunner [atlassian- diagnostics-data- logger] 1591241235004 ; INFO ; DB ; DB-2001 ; Slow event listener detected ;</pre>	<p>An event was successfully dispatched to an event listener, but the event or listener took a long time to process it. For synchronous events, the user request that triggered the event had to wait a long time for the request to complete. For asynchronous events, one of the event processing threads was unavailable to dispatch other events during this time.</p>
<pre>INFO ServiceRunner [atlassian- diagnostics-data- logger] 1591241235004 ; INFO ; DB ; DB-3001 ; Slow HTTP request detected ;</pre>	<p>An HTTP request took longer than 60 seconds to complete.</p>

 3-rd party apps can also generate custom alerts that are visible in the logs.

## Retention and defaults

Some behaviours trigger a single alert, for others, multiple alerts are possible. Diagnostic information is stored in the database, and retained for 30 days. Old alerts are cleaned up automatically.

### Change the property default values

Upping the values will result in fewer alerts to be triggered.

- The values can be modified either from the **\$JIRA\_HOME/jira-config.properties**. For example,

```
jira.diagnostics.thresholds.slow-query-millis=5000  
jira.diagnostics.thresholds.number-of-results=10000
```

- They can also be modified using the following JVM parameters:

```
-Djira.diagnostics.thresholds.slow-query-millis=5000  
-Djira.diagnostics.thresholds.number-of-results=10000
```

Make sure there are no spaces between the equal sign in both methods.

#### **Change the default retention**

To change the default retention of 30 days, do the following:

1. Open **jira-config.properties**.
2. Edit the following metric changing the number of retention days:`com.atlassian.jira.health.diagnostics.alerts.retention-period-days: <number of days to retain Diagnostics Alerts>`

# Viewing Jira application instrumentation statistics

Jira provides an **Instrumentation** page, which displays a variety of statistics on a wide range of internal properties within Jira that have been 'instrumented' (i.e. recorded) for presentation through Jira's administration area.

This page is mostly useful to help Atlassian Support provide assistance with your support queries, especially if they ask you to quote the statistics of one or more properties listed on this page.

**Note:** For all of the following procedures, you must be logged in as a user with the **Jira Administrators** [global permission](#).

1. In the upper-right corner of the screen, select **Administration**  **> System**.

2. Select **System support > Instrumentation** to display the Instrumentation page.

Name	Type	Value	Invocation	Time (ms)	CPU (nanos)
cache.CachingFieldConfigContextPersister.evictionCount	Counter	0			
cache.CachingFieldConfigContextPersister.hitCount	Counter	1,314			
cache.CachingFieldConfigContextPersister.loadExceptionCount	Counter	0			
cache.CachingFieldConfigContextPersister.loadSuccessCount	Counter	6			
cache.CachingFieldConfigContextPersister.missCount	Counter	6			
cache.CachingFieldConfigContextPersister.size	Gauge	6			
cache.CachingFieldConfigContextPersister.totalLoadTime	Counter	5			
cache.DefaultFieldLayoutManager.evictionCount	Counter	0			
cache.DefaultFieldLayoutManager.hitCount	Counter	14,533			
cache.DefaultFieldLayoutManager.loadExceptionCount	Counter	0			
cache.DefaultFieldLayoutManager.loadSuccessCount	Counter	3			
cache.DefaultFieldLayoutManager.missCount	Counter	4			
cache.DefaultFieldLayoutManager.size	Gauge	1			
cache.DefaultFieldLayoutManager.totalLoadTime	Counter	12			
cache.DefaultIssueLinkManager.evictionCount	Counter	0			
cache.DefaultIssueLinkManager.hitCount	Counter	1,935			
cache.DefaultIssueLinkManager.loadExceptionCount	Counter	0			
cache.DefaultIssueLinkManager.loadSuccessCount	Counter	602			
cache.DefaultIssueLinkManager.missCount	Counter	602			
cache.DefaultIssueLinkManager.size	Gauge	602			
cache.DefaultIssueLinkManager.totalLoadTime	Counter	4,701			
cache.DefaultPermissionSchemeManager.evictionCount	Counter	0			
cache.DefaultPermissionSchemeManager.hitCount	Counter	17,824			
cache.DefaultPermissionSchemeManager.loadExceptionCount	Counter	0			
cache.DefaultPermissionSchemeManager.loadSuccessCount	Counter	2			
cache.DefaultPermissionSchemeManager.missCount	Counter	2			
cache.DefaultPermissionSchemeManager.size	Gauge	1			
cache.DefaultPermissionSchemeManager.totalLoadTime	Counter	6			
cache.DefaultUserPropertyManager.evictionCount	Counter	38			
cache.DefaultUserPropertyManager.hitCount	Counter	713,077			
cache.DefaultUserPropertyManager.loadExceptionCount	Counter	0			
cache.DefaultUserPropertyManager.loadSuccessCount	Counter	43			
cache.DefaultUserPropertyManager.missCount	Counter	43			
cache.DefaultUserPropertyManager.size	Gauge	5			
cache.DefaultUserPropertyManager.totalLoadTime	Counter	17			
cache.JiraOsgiContainerManager.evictionCount	Counter	80			
cache.JiraOsgiContainerManager.hitCount	Counter	34,914			
cache.JiraOsgiContainerManager.loadExceptionCount	Counter	0			
cache.JiraOsgiContainerManager.loadSuccessCount	Counter	86			
cache.JiraOsgiContainerManager.missCount	Counter	86			

cache.JiraOsgiContainerManager.size	Gauge	6			
cache.JiraOsgiContainerManager.totalLoadTime	Counter	134			
cache.VelocityTemplateCache.directives.evictionCount	Counter	0			
cache.VelocityTemplateCache.directives.hitCount	Counter	419,353			
cache.VelocityTemplateCache.directives.loadExceptionCount	Counter	0			
cache.VelocityTemplateCache.directives.loadSuccessCount	Counter	76			
cache.VelocityTemplateCache.directives.missCount	Counter	76			
cache.VelocityTemplateCache.directives.size	Gauge	76			
cache.VelocityTemplateCache.directives.totalLoadTime	Counter	0			
cache.VelocityTemplateCache.evictionCount	Counter	0			
cache.VelocityTemplateCache.hitCount	Counter	419,353			
cache.VelocityTemplateCache.loadExceptionCount	Counter	0			
cache.VelocityTemplateCache.loadSuccessCount	Counter	76			
cache.VelocityTemplateCache.missCount	Counter	76			
cache.VelocityTemplateCache.size	Gauge	76			
cache.VelocityTemplateCache.totalLoadTime	Counter	10			
concurrent.users	Gauge	1			
db.conns	Operation		112,352	1,058,122,695	0
db.conns.borrowed	Gauge	1			
db.reads	Operation		103,234	2,238	0
db.writes	Operation		4,669	406	0
dbcp.maxActive	Gauge	20			
dbcp.numActive	Gauge	1			
dbcp.numIdle	Gauge	19			
entity.customfields.total	Gauge	1			
entity.groups.total	Gauge				
entity.issues.total	Gauge	301			
entity.projects.total	Gauge	2			
entity.users.total	Gauge				
entity.workflows.total	Gauge	4			
http.session.objects	Gauge	10			
http.sessions	Gauge	2			
index.writes	Operation		9	13,829	0
issue.index.reads	Operation		96	653	0
jmx.class.loaded.current	Gauge	26,029			
jmx.class.loaded.total	Counter	27,594			
jmx.class.unloaded.total	Counter	1,565			
jmx.gc	Operation		194	25,087	0
jmx.memory.heap.committed	Gauge	659,767,296			
jmx.memory.heap.used	Gauge	564,776,504			
jmx.memory.nonheap.committed	Gauge	146,669,568			
jmx.memory.nonheap.used	Gauge	146,505,104			

jmx.system.up.time	Gauge	1,058,358,664		
jmx.thread.cpu.block.count	Counter	0		
jmx.thread.cpu.block.time	Counter	0		
jmx.thread.cpu.time	Counter	323,250,000,000		
jmx.thread.cpu.user.time	Counter	259,125,000,000		
jmx.thread.cpu.wait.count	Counter	0		
jmx.thread.cpu.wait.time	Counter	0		
jmx.thread.daemon.count	Gauge	32		
jmx.thread.ever.count	Gauge	4,445		
jmx.thread.nondaemon.count	Gauge	11		
jmx.thread.peak.count	Gauge	47		
jmx.thread.total.count	Gauge	43		
searcher.lucene.close	Counter	10		
searcher.lucene.open	Counter	13		
web.requests	Operation	4,328	623,512	0

**JMX Support Information**

- Thread Contention Monitoring - Supported - Not Enabled
- Thread CPU Time Monitoring - Supported - Not Enabled

# Generating a thread dump

Occasionally, Jira may appear to 'freeze' during execution of an operation. During these times, it is helpful to retrieve a **thread dump** a log containing information about currently running threads and processes within the Java Virtual Machine. Taking thread-dumps is a non-destructive process that can be run on live systems. This document describes the steps necessary to retrieve a **thread dump**.

The steps necessary to retrieve the **thread dump** are dependant on the operating system Jira is running in please follow the appropriate steps below.

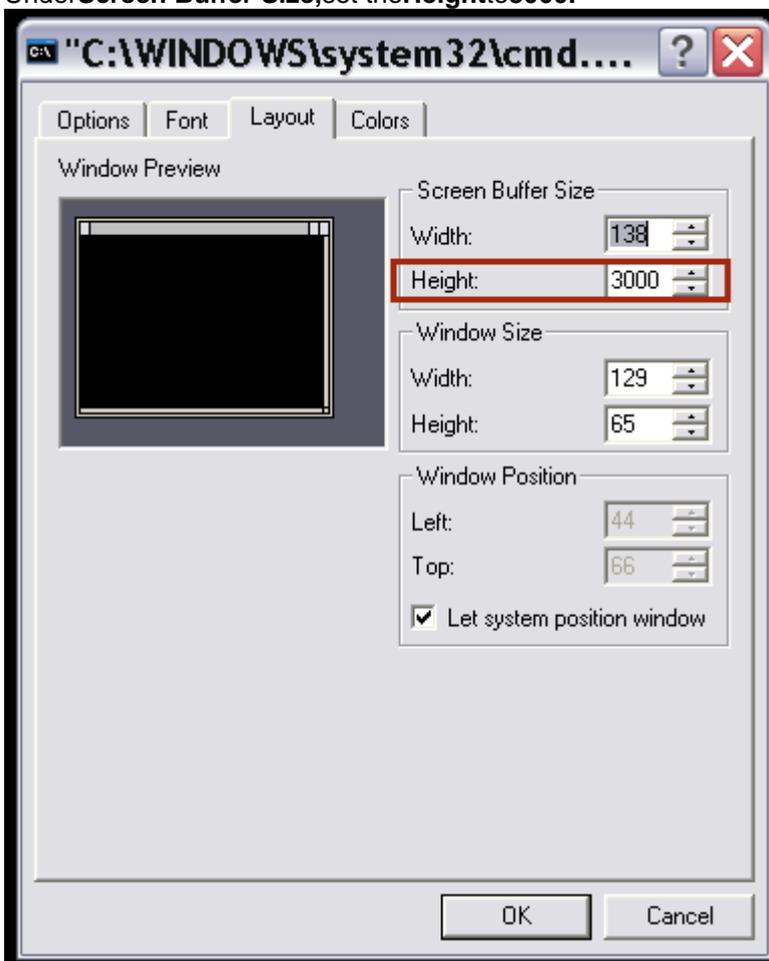
## On this page:

- [Windows environment](#)
- [Linux/Unix/OS X environment](#)
- [Analysis tools](#)

## Windows environment

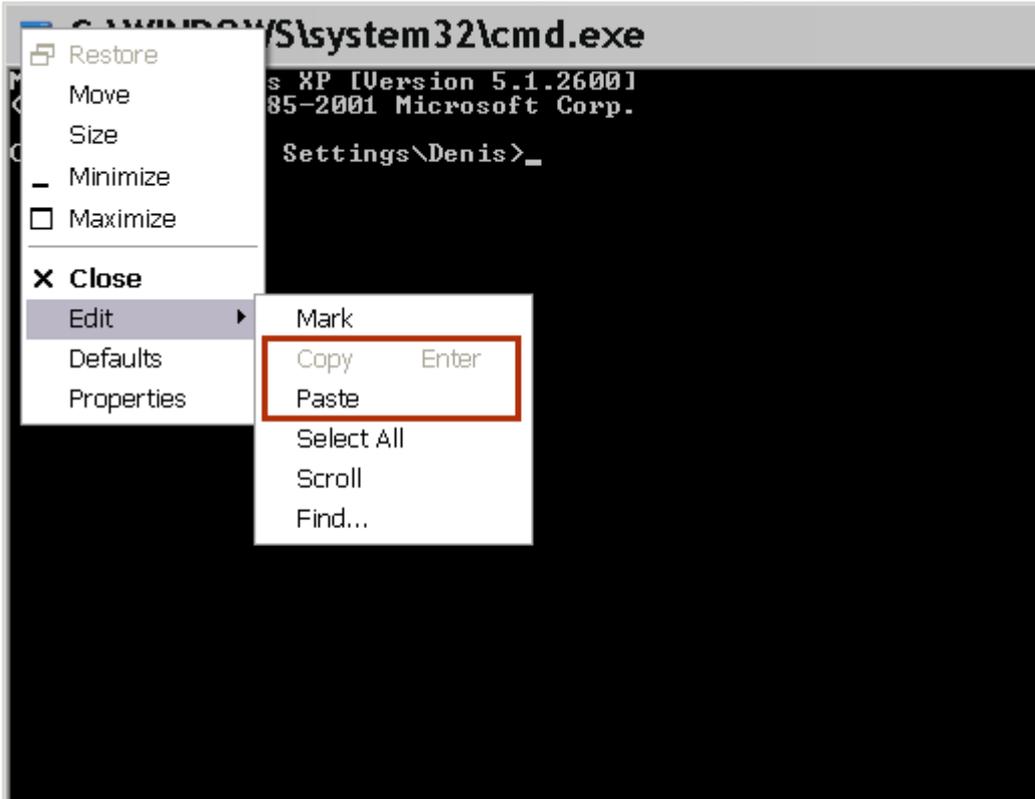
### Jira running from startup.bat

1. In the **Command Console** window where Jira is running, open the properties dialog box by right clicking on the title bar and select "**Properties**".
2. Select the **Layout** tab.
3. Under **Screen Buffer Size**, set the **Height** to **3000**.



4. Click **OK**.
5. With the same command console in focus, press **CTRL-BREAK**. This will output the thread dump to the command console.
6. Scroll back in the command console until you reach the line containing "Full thread dump".
7. Right click the title bar and select **Edit -> Mark**. Highlight the entire text of the thread dump.

8. Right click the title bar and select **Edit -> Copy**. The thread dump can then be pasted into a text file.



## Jira running as a Windows service

### Using jstack

The JDK ships with a tool named `jstack` for generating thread dumps.

1. Identify the process. Launch the task manager by, pressing `Ctrl + Shift + Esc` and find the Process ID of the Java (Jira) process. You may need to add the PID column using `View->Select Columns ...`.
2. Run `jstack` to capture a single thread dump or multiple thread dumps at set intervals:

Use the following command to capture a single thread dump of the process ID `<JIRA_PID>`:

```
jstack.exe -l <JIRA_PID> > threaddump.txt
```

For example, if the PID is 22668, enter:

```
jstack.exe -l 22668 > threaddump.txt
```

This will output to `threaddump.txt` in your current directory.

Use the following command to capture 6 thread dumps of the process id `<JIRA_PID>` in 10-second intervals between each thread dump:

```
for /L %n in (1,1,6) do timeout 10 | jstack.exe -l <JIRA_PID> > threaddump-%n.txt
```

For example, if the PID is 22668, enter:

```
for /L %n in (1,1,6) do timeout 10 | jstack.exe -l 22668 > threaddump-%n.txt
```

This will output to `threaddump-%n.txt`, where `%n` is the number of the loop iteration (starting at 1).

**i** You can modify the `timeout` command parameter to adjust the time between each thread dump and the range in the `in (start,step,end)` clause to adjust the number of thread dumps to capture. The `(1,1,6)` range in the example above means the following:

- Start at 1
- Increment by 1
- End at 6

**w** Common issues with `jstack`:

- You must run `jstack` as the same user that is running Jira.
- If you get the error "Not enough storage is available to process this command", download the 'psexec' utility from [here](#), and then run one of the following commands, where `<JIRA_PID>` is the Jira process ID (for example, 22668):

- To capture a single thread dump:

```
jstack.exe -l <JIRA_PID> > threaddump.txt
```

- To capture multiple thread dumps at set intervals:

```
for /L %n in (1,1,6) do timeout 10 | psexec -s jstack.exe -l $JIRA_PID > threaddump-%n.txt
```

- If the `jstack` executable is not in your `$PATH`, then look for it in your `<JDK_HOME>/bin` directory
- If you receive `java.lang.NoClassDefFoundError: sun/tools/jstack/JStack` check that `tools.jar` is present in your JDK's lib directory. If it is not, download a full version of the JDK.

## Linux/Unix/OS X environment

### Linux/Unix command line

1. Identify the `java` process that Jira is running in. This can be achieved by running a command similar to:

```
ps -ef | grep java
```

The process will appear similarly as follows:

```
keithb      910    873    1 17:01 pts/3      00:00:18 /usr/java/jdk/bin/java -Xms128m -Xmx256m
-Xms128m -Xmx256m -Djava.awt.headless=true -Djava.util.logging.manager=org.apache.juli.
ClassLoaderLogManager
-Djava.awt.headless=true -Djava.endorsed.dirs=/tmp/atlassian-jira-enterprise-3.6-standalone/common
/endorsed
-classpath :
```

2. In order to retrieve the thread dump, execute the command

```
kill -3 <pid>
```

where `pid` is the process id in this case, 910.

3. The thread dump will be written to the Tomcat console output. The console output is redirected to the `logs/catalina.outfile`, which can be found in the [Jira application installation directory](#) for JIRA Standalone / Installer.

### Linux/Unix Alternative: Generating thread dumps using `jstack`

If you have trouble using `kill -3 <pid>` to obtain a thread dump, try using `jstack` a java utility that will output stack traces of Java threads for a given process.

1. Identify the **java** process that Jira is running in. This can be achieved by running a command similar to:

```
ps -ef | grep java
```

The process will appear similarly as follows:

```
adam 22668 0.3 14.9 1691788 903928 ? S1 Jan27 9:36 /usr/lib/jvm/java-6-sun-1.6.0.14/bin/java -
Djava.util.logging.config.file=/home/adam/Products/installs/atlassian-jira-enterprise-4.0.1-
standalone/conf/logging.properties -XX:MaxPermSize=256m -Xms128m -Xmx1048m -Djava.awt.
headless=true -Datlassian.standalone=JIRA -Dorg.apache.jasper.runtime.BodyContentImpl.
LIMIT_BUFFER=true -Dmail.mime.decodeparameters=true -Datlassian.mail.senddisabled=false -
Datlassian.mail.fetchdisabled=false -Djava.util.logging.manager=org.apache.juli.
ClassLoaderLogManager -Djava.endorsed.dirs=/home/adam/Products/installs/atlassian-jira-enterprise-
4.0.1-standalone/common/endorsed -classpath /home/adam/Products/installs/atlassian-jira-
enterprise-4.0.1-standalone/bin/bootstrap.jar -Dcatalina.base=/home/adam/Products/installs
/atlassian-jira-enterprise-4.0.1-standalone -Dcatalina.home=/home/adam/Products/installs
/atlassian-jira-enterprise-4.0.1-standalone -Djava.io.tmpdir=/home/adam/Products/installs
/atlassian-jira-enterprise-4.0.1-standalone/temp org.apache.catalina.startup.Bootstrap start
```

2. Run `jstack` to capture a single thread dump or multiple thread dumps at set intervals:

Use the following command to capture a single thread dump of the process ID `<JIRA_PID>`:

```
jstack <JIRA_PID> > threaddump.txt
```

For example, if the PID is 22668, enter:

```
jstack 22668 > threaddump.txt
```

The output will be saved to `threaddump.txt` in the current directory.

Use the following command to capture 6 thread dumps of the process ID `<JIRA_PID>` in 10-second intervals between each thread dump:

```
for i in $(seq 6); do top -b -H -p <JIRA_PID> -n 1 > threaddump.$(date +%s).txt; kill -3
<JIRA_PID>; sleep 10; done
```

For example, if the PID is 22668, enter:

```
for i in $(seq 6); do top -b -H -p 22668 -n 1 > threaddump.$(date +%s).txt; kill -3 22668;
sleep 10; done
```

The output will be saved to `threaddump.$(date +%s).txt` in the current directory, where `date +%s` is the current Unix timestamp.

 You can modify the `seq` command parameter to adjust the the number of thread dumps to capture and the `sleep` command parameter to adjust the time between each thread dump.

 If you are connecting to the server through RDP, `jstack` might fail with following error:

```
Not enough storage is available to process this command
```

You will need to open a RDP session in console mode: `mstsc /admin`

## Linux/Unix Alternative: Generating thread dumps using scripts

You can also generate a thread dump by using scripts prepared by our Support team. That's an easy process, and the scripts will do everything for you. As an addition to generating a thread dump, the scripts also allow you to generate heap dumps, check the disk access speed, or the Java SSL connection.

1. (Optional) Install the [Thready](#) app if it's supported by your Jira version. Thready renames Tomcat threads to include the URL they serve, which will make it easier to analyze the threads.
2. Download and install the scripts from <https://bitbucket.org/atlassianlabs/atlassian-support/>.
3. Execute the scripts when your Jira instance behaves slowly or is unresponsive.
  - a. (Optional) The scripts also allow you to test the disk access speed. This is described in more detail in [Testing disk access speed](#).
  - b. When asked whether you want to capture thread dumps, enter **Y**.
  - c. When asked whether you want to capture heap dumps, enter **N**.
  - d. (Optional) The scripts also allow you to check the Java SSL connection.
4. After running the scripts, the thread dump will be captured and compressed. You can now open a Support ticket and attach the generated package.

## Analysis tools

Try [TDA](#) or [Samurai](#) to inspect your thread dump.

### TDA

1. Download [TDA](#).
2. CD to the directory where the JAR exists.
3. Run:

```
java -jar -Xmx512M ~/tda-bin-1.6/tda.jar
```

4. Open your catalina.out file, containing the thread dump.

### Issue processing thread dump with TDA (NumberFormatException)

Should you get an error on TDA console like:

```
Exception in thread "AWT-EventQueue-0" java.lang.NumberFormatException: For input string: "5 os_prio=0"
    at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
    at java.lang.Long.parseLong(Long.java:441)
    at java.lang.Long.<init>(Long.java:702)
    at com.pironet.tda.utils.ThreadsTableModel.getValueAt(ThreadsTableModel.java:80)
    at com.pironet.tda.utils.TableSorter.getValueAt(TableSorter.java:285)
    at javax.swing.JTable.getValueAt(JTable.java:2717)
    at javax.swing.JTable.prepareRenderer(JTable.java:5719)
    at javax.swing.plaf.basic.BasicTableUI.paintCell(BasicTableUI.java:2114)
    at javax.swing.plaf.basic.BasicTableUI.paintCells(BasicTableUI.java:2016)
    at javax.swing.plaf.basic.BasicTableUI.paint(BasicTableUI.java:1812)
    at javax.swing.plaf.ComponentUI.update(ComponentUI.java:161)
    at javax.swing.JComponent.paintComponent(JComponent.java:778)
    at javax.swing.JComponent.paint(JComponent.java:1054)
    at javax.swing.JComponent.paintChildren(JComponent.java:887)
    at javax.swing.JComponent.paint(JComponent.java:1063)
    at javax.swing.JViewport.paint(JViewport.java:731)
    at javax.swing.JComponent.paintChildren(JComponent.java:887)
    at javax.swing.JComponent.paint(JComponent.java:1063)
    at javax.swing.JComponent.paintChildren(JComponent.java:887)
    at javax.swing.JSplitPane.paintChildren(JSplitPane.java:1047)
    at javax.swing.JComponent.paint(JComponent.java:1063)
    at javax.swing.JComponent.paintToOffscreen(JComponent.java:5230)
    at javax.swing.BufferStrategyPaintManager.paint(BufferStrategyPaintManager.java:295)
    at javax.swing.RepaintManager.paint(RepaintManager.java:1249)
    at javax.swing.JComponent._paintImmediately(JComponent.java:5178)
    at javax.swing.JComponent.paintImmediately(JComponent.java:4989)
    at javax.swing.RepaintManager$3.run(RepaintManager.java:808)
    at javax.swing.RepaintManager$3.run(RepaintManager.java:796)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.security.ProtectionDomain$1.doIntersectionPrivilege(ProtectionDomain.java:76)
```

```

at javax.swing.RepaintManager.paintDirtyRegions(RepaintManager.java:796)
at javax.swing.RepaintManager.paintDirtyRegions(RepaintManager.java:769)
at javax.swing.RepaintManager.prePaintDirtyRegions(RepaintManager.java:718)
at javax.swing.RepaintManager.access$1100(RepaintManager.java:62)
at javax.swing.RepaintManager$ProcessingRunnable.run(RepaintManager.java:1677)
at java.awt.event.InvocationEvent.dispatch(InvocationEvent.java:312)
at java.awt.EventQueue.dispatchEventImpl(EventQueue.java:733)
at java.awt.EventQueue.access$200(EventQueue.java:103)
at java.awt.EventQueue$3.run(EventQueue.java:694)
at java.awt.EventQueue$3.run(EventQueue.java:692)
at java.security.AccessController.doPrivileged(Native Method)
at java.security.ProtectionDomain$1.doIntersectionPrivilege(ProtectionDomain.java:76)
at java.awt.EventQueue.dispatchEvent(EventQueue.java:703)
at java.awt.EventDispatchThread.pumpOneEventForFilters(EventDispatchThread.java:242)
at java.awt.EventDispatchThread.pumpEventsForFilter(EventDispatchThread.java:161)
at java.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.java:150)
at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:146)
at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:138)
at java.awt.EventDispatchThread.run(EventDispatchThread.java:91)

```

Apply the following command on the thread dump(s) to fix the thread header format to make it processable:

```
sed -i 's/prio=[0-9]\{1,2\} os_prio=[0-9]\{1,2\}/prio=5/g' <filename>
```

```
sed -i " 's/prio=[0-9]\{1,2\} os_prio=[0-9]\{1,2\}/prio=5/g' <filename>
```

**Check the known thread dump knowledge base articles:**

- [Troubleshoot index problems in Jira server](#)
- [Jira slow with dangerous use of multiple connections error in log](#)
- [JIRA Deadlocks when Running Tomcat 6.0.24](#)
- [OutOfMemory or Poor Performance due to XML View of a Filter](#)
- [\(Archived\) JIRA applications performance tuning](#)
- [Jira server crashes with OutofMemory Java heap space error](#)

# Finding your Jira application Support Entitlement Number (SEN)

There are three ways to find your Support Entitlement Number (SEN).

See [How to find your Support Entitlement Number \(SEN\)](#) in the support space for more general information about how Atlassian Support uses this number.

## Method 1: Check in the Jira administration interface

Access the Jira license page, as described in [Licensing your Jira applications](#). The Jira license page will show your Support Entitlement Number (SEN).

**License Information**

This page shows your current licensing information. You can use the Update License form to update the license JIRA is running with.

Organization	dailyplanet
Date Purchased	16/Nov/11
License Type	(Support and updates available until 15/Nov/11)
Server ID	BKPQ-MSIU-F3AD-U6LF
Support Entitlement Number (SEN)	SEN-500
User Limit	Unlimited

**Update License**

Copy and paste the license key below. You can access your license key on [My Account](#).

License

## Method 2: Check [my.atlassian.com](http://my.atlassian.com)

Your Support Entitlement Number is available from the licenses page after logging in to <http://my.atlassian.com>:

**Welcome**

Hey [redacted], welcome to the Atlassian customer portal. For questions relating to [managing your account](#), licensing or purchasing, please see the [purchasing and licensing FAQ](#) or [contact a customer service representative](#). To request technical support, please visit our [Support Portal](#).

**Licenses** [New Evaluation License](#) | [New SourceTree License](#)

You do not have any product licenses.

**Evaluations** [New Evaluation License](#) | [Expand All](#) | [Collapse All](#)

Product	Name	Support Expires	Support
<input type="checkbox"/> JIRA (Unlimited Users): Evaluation	Atlassian	14 Dec [redacted]	

Server ID: BCQU-[redacted] License Key: [redacted]

SEN [redacted]

**This license is compatible with JIRA 4 or above.**

Actions: [Buy](#) | [Download JIRA](#)

## Method 3: Check your Atlassian invoice

Your Support Entitlement Number (SEN) also appears on the third page of your Atlassian Invoice.

# Auditing in Jira

The auditing feature tracks key activities in Jira products. These activities are recorded in an audit log that can be viewed in the Jira administration console. This can be a handy tool in helping you diagnose problems in Jira products or used for security and compliance purposes.

The feature is available in both Server and Data Center, however its scope is more extensive in Data Center. To view the full audit log you need to be a Jira System admin or to have a **Jira Administrator** global permission.

Data Center admins can also allow project admins to have access the audit log for a specific project. Any project admin needs to have the Administer project and the Browse projects permissions.

**i** To restrict the audit log to system administrators only set the `plugin.audit.log.view.sysadmin.only` property to `true`. Then you can prevent people with Jira Administrator global permission from accessing the audit log.

## On this page:

- [Main differences between auditing in Server and Data Center](#)
- [View the audit log](#)
- [Search and filter the audit log](#)
- [Edit log settings](#)
- [Export the audit log](#)
- [Access the audit log file DATA CENTER](#)
- [Integrate with external software DATA CENTER](#)
- [Audit log and migration](#)
- [Auditing and the REST API](#)
- [Auditing properties](#)

## Main differences between auditing in Server and Data Center

Functionality	Jira Server	Jira Data Center
Coverage areas	✓ (fewer than in Data Center)	✓
Selecting coverage levels	✗ (Only Base and Off are available.)	✓
Setting DB log retention	✓	✓
Setting file retention	✗	✓
Storing log file in two locations	✗	✓
3rd-party integrations	✗	✓
Exporting latest 100k results	✓	✓
Exporting filtered results	✗	✓
Audit log for project admins	✗	✓

## View the audit log

To view the events in the audit log:

1. From the app header, select **Administration**  > **System**
2. From the left menu, select **Audit log**
3. Click on each event to expand it and see the details.

Apr 06, 2020, 09:23:05 AM GMT+2	projects	Project roles changed	Administrators	Server Services Team
Source:	10.125.96.121			
Node ID:	node1			
Method:	Browser			
Users:	+ JIRAUSER174716			

Information for each event may include:

- **Source**- IP address of the user who performed the action (though not recorded for system-generated events). Can also show the node IP address.
- **Node ID**- unique ID of the node where the action was performed.
- **Method**- depending on how the action was performed, will be either Browser (end user) or System (system process).

Admins, system admins, and project admins can also access audit logs for specific projects. A project-specific log can be found in **Project settings** and contains events relating to a this particular project.

**i** Sprint Deleted events are recorded only in the system audit log. This is because sprints are global objects that aren't tied to a particular project or board.

## Search and filter the audit log

You can search the event on keyword and filter by:

- Category **DATA CENTER**
- Summary **DATA CENTER**

**i** Click **More** to see the filters. Note that the category and the summary filters are not inclusive.

- User
- Project
- Date (you need to provide date and time)

Your query can be up to 100 characters long. To speed things up, we initially search 1 million events. After this search is performed, you have an option to run a full database search. Note that the full search might take a while.

## Edit log settings

In the audit log settings you can decide how long you want to retain the logged events in the database and the areas from which you want to collect the logs.

### Update database retention

The database retention is limited by the retention period and the default cap of 10 million records. The cap is configurable using the `plugin.audit.db.limit.rows` property. If you decide to modify it, make sure your database is big enough to store all the events.

To update your database retention period, do the following:

1. Select **Actions > Settings**.
2. Update the retention period.
3. Click **Save**.

**i** If you choose a long retention period, it might affect the size and performance of your database. If you decide to lower the period, all the events that exceed the newly set period will be deleted and disappear from the page. You might consider creating a backup before you lower the retention period.

If you migrated from a previous Jira version, your default retention period is 20 years. If you have a new Jira installation, its 3 years.

For more on selecting an optimal retention period, see [How do I know what retention period is good for my audit log?](#)

## Update log file retention

In Jira Data Center, we write events to an audit log file in your local home directory. This file can be used as an additional record, and when integrating with third-party logging aggregation tools.

The maximum size of these files is 100 MB, so make sure you've provisioned enough disk space on each node, especially if you have set the logging level to Advanced or Full.

1. Select **Actions > Settings**.
2. Update the number of files you want to store.
3. Click **Save**.

Once a node reaches the log file retention limit, the oldest one is deleted. If you need to keep these logs, for example for compliance purposes, you may want to manually back up the files in this directory on a regular basis, or send them to a third party logging platform. See [Audit log Integrations in Jira](#).

## Select events to log

The events that are logged are organized in categories that belong to specific coverage areas. For example, login-related events are in the Login category that belongs to the Security coverage area.

Date	Author	Category	Summary	Affected object
> Feb 18, 2020, 02:08:00 AM UTC	Anonymous	login	User failed to log in	

For all coverage areas and events logged in each area, see [Audit log events in Jira](#).

To adjust the coverage:

1. Go to **> Settings**.
2. In the **Coverage level** drop-down, choose the level to log the events you need or **Off** to stop collecting events from a particular area.

Coverage levels reflect the number and frequency of events that are logged.

- **Off:** Turns off logging events from this coverage area.
- **Base:** Logs low-frequency and some of the high-frequency core events from selected coverage areas.

Available only in Data Center:

- **Advanced:** Logs the core events as well as the low and medium frequency events from the coverage areas that are available only for Data Center.
- **Full:** Logs all the events available in Base and Advanced, plus additional events for a comprehensive audit.

**i** Changing coverage level changes the set of the events that are logged. If you can't find a specific event, it might be because coverage level was changed and these events were not logged for a period of time. Check all audit log configuration events to see if that might have been the case.

## Export the audit log

You can export up to 100,000 events as a CSV file. If you have more than 100,000 events, only the 100,000 newest events are included in the export. If you run Jira Data Center, you can also export filtered results up to 100,000 events.

1. Navigate to the Audit log/ Advanced audit log and choose **Export**.
2. **DATA CENTER** Select to export the latest 100,000 or filtered results.
3. Confirm by clicking **Export** again.

## Access the audit log file **DATA CENTER**

For Jira Data Center, each node has its own log and can be found in the <Jira localhome directory>/log/audit directory. The log is stored as a JSON file.

This directory has a file limit of 100 files, and each file has a size limit of 100 MB. Jira checks the directory every 24hrs. If these limits have been reached, it will delete the oldest file.

## Integrate with external software **DATA CENTER**

You can use the log file to integrate with ELK, Splunk, Sumologic, and Amazon CloudWatch. For more information on integrations, see [Audit log integrations in Jira](#).

## Audit log and migration

### Migrate database

If you have more than 10 million events stored in your database, and you move to a new database, only the latest 10 million will be migrated and the remaining data will be removed. To have access to your older events you can either create a backup before you migrate and access the data in the backup.

### Migrate from a previous Jira version

Migrating audit log records might take up to several hours depending on the size of the audit log and the type of your database. If you migrate using ZDU, you can still use your Jira during migration.

You can use the `jira.advanced.audit.log.migration.limitflag` in the Jira properties file to limit the number of events you want to migrate or to turn off the migration altogether. If you decide to turn off migration, your new audit log will only show the events that happen after upgrade.

- ✔ To make sure the limit works, provide the property directly as a command line argument or via the `setenv.sh` or `setenv.batfile`. For example:

```
JVM_SUPPORT_RECOMMENDED_ARGS="-Djira.advanced.audit.log.migration.limit=10"
```

- ⓘ We migrate the existing events to the database not to files. We migrate only 10 million entries from the old audit table to the new one.

## Auditing and the REST API

The audit log can also be accessed via the [REST API](#).

### Auditing properties

These properties control the auditing feature, determining the number of audit entries logged, or stored in the database, and the size of those entries. Changing these settings will only affect new audit entries.

Increasing the amount of auditing done may have an adverse effect on performance.

Default value	Description
	plugin.audit.search.max.concurrent.nontext.requests
10	Maximum number of concurrent non-freetext search requests allowed, defaults to 10 per node
	plugin.audit.search.max.concurrent.text.requests
5	Maximum number of concurrent freetext search requests allowed, defaults to 5 per node
	plugin.audit.search.query.timeout
30	Timeout in seconds for a queued search request, defaults to 30 seconds
	plugin.audit.db.limit.rows
10000000	Maximum number of audit event rows stored in DB, events exceeding the limit get deleted in time order, defaults to 10M checked on hourly basis
	plugin.audit.db.limit.buffer.rows
1000	Buffer to accommodate new audit events, defaults to 1000 rows
	plugin.audit.db.delete.batch.limit
10000	maximum number of events to be deleted per database transaction used when enforcing retention limits, defaults to 10,000 rows
	plugin.audit.schedule.db.limiter.interval.mins
60	Database size check, running every 60 minutes
	plugin.audit.broker.exception.loggedCount
3	Maximum number of audit events written to system log file in case of error, defaults to 3
	plugin.audit.retention.interval.hours
24	Database retention check, which deletes events exceeding retention period, running every 24 hours
	plugin.audit.file.max.file.size
100	Size limit in megabytes for individual audit file, file rotates when limit is reached, defaults to 100MB
	plugin.audit.file.max.file.count
100	Maximum number of audit files, the earliest file will be deleted when limit is reached, defaults to 100
	plugin.audit.consumer.buffer.size
10000	Maximum number of audit events kept in buffer waiting to be consumed, defaults to 10,000
	plugin.audit.broker.default.batch.size
3000	Maximum number of audit events dispatched to consumer, defaults to 3,000 per batch
	plugin.audit.coverage.cache.read.expiration.seconds
30	How long the coverage cache is valid, defaults to 30 seconds

# Audit log events in Jira

Refer to the tables below for more details about each Jira audit log events. Note that some coverage areas are available in Data Center only.

Each table represents a specific coverage area, and groups all events by coverage level and category.

 The events generated by external apps that call Jira REST API that fall into the Apps coverage area are not listed here because they are app-dependent.

## On this page:

- [Global configuration and administration](#)
- [User management](#)
- [Permissions](#)
- [Local configuration and administration](#)
- [Security](#)
- [End user activity](#)

## Global configuration and administration

### Category: User interface

Coverage level	Events logged
Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	Announcement banner updated, Default user settings changed
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS LOGGED HERE

### Category: Issue types

Coverage level	Events logged
Base	Issue type created
Advanced (additional events on top of Base)	Issue type scheme created, Issue type scheme updated, Issue type scheme deleted, Association of issue type scheme changed, Issue type updated, Issue type deleted, Issue type sub-task created, Issue type sub-task updated, Issue type sub-task deleted, Sub-tasks enabled, Sub-tasks disabled, Association of issue type scheme changed, Sub-tasks issue type disabled, Sub-tasks issue type disabled, Sub-task issue type updated, Sub-task issue type deleted
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

### Category: Screens

Coverage level	Events logged
Base	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

Advanced (additional events on top of Base)	Issue type screen scheme created, Issue type screen scheme updated, Issue type screen scheme deleted, Issue type screen scheme removed from screen, Issue type screen scheme associated, Screen scheme created, Screen scheme updated, Screen scheme deleted, Screen scheme copied, Screen associated with screen scheme, Screen updated for screen scheme, Screen removed from screen scheme
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

**Category: General configuration**

Coverage level	Events logged
Base	Velocity Chart: Max. number of sprints updated, Velocity Chart: Max. number of issues updated
Advanced (additional events on top of Base)	Time tracking enabled, Time tracking disabled Jira Service Management: Public signup enabled, Public signup disabled
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

**Category: Projects**

Coverage level	Events logged
Base	Project category updated, Project role updated, Project role deleted
Advanced (additional events on top of Base)	Project role created, Project role deleted, Project role changed
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

**Category: System**

Coverage level	Events logged
Base	Upgrade started, Upgrade approved, Upgrade failed, Upgrade canceled, Upgrade finished, Application property modified
Advanced (additional events on top of Base)	Dark feature enabled, Dark feature disabled JMX monitoring enabled, JMX monitoring disabled, Jira service deleted
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

**Category: Mail settings**

Coverage level	Events logged
Base	NO EVENTS LOGGED HERE

Advanced (additional events on top of Base)	Outgoing mail enabled, Outgoing mail disabled, Email queue flushing requested
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

**Category: Fields**

Coverage level	Events logged
Base	Field configuration scheme created, Field configuration scheme updated, Field configuration scheme deleted, Field configuration scheme copied, Custom field created, Custom field deleted, Custom field updated, Field configuration created, Field configuration created, Field configuration removed, Field configuration updated, Field configuration scheme added to project, Field configuration scheme removed from project, Field added, Field moved, Field deleted, Tab added, Tab moved, Tab removed, Tab renamed
Advanced (additional events on top of Base)	Custom field context added, Custom field context modified, Custom field context deleted, Default resolution updated, Resolution deleted, Resolution updated, New resolution created, Status updated (8.11.0 +), Status order changed (8.11.0 +), Status deleted (8.11.0 +), Status created (8.11.0 +), Priority created (8.12.0 +), Priority updated (8.12.0 +), Priority deleted (8.12.0 +)
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

**Category: Indexing**

Coverage level	Events logged
Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	Full re-index started, Full re-index completed, Background re-index started, Background re-index completed, Background re-index canceled
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

**Category: Priorities**

Coverage level	Events logged
Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	Priority scheme created, Priority scheme updated, Priority scheme deleted, Priority scheme unassigned from project, Priority scheme assigned to project
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

**Category: Field config scheme**

Coverage level	Events logged
----------------	---------------

Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	Priority scheme association changed
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

**Category: Permissions**

Coverage level	Events logged
Base	Permission scheme created, Permission scheme copied, Permission scheme deleted, Permission scheme updated, Permission added, Permission deleted, Global permission added, Global permission deleted
Advanced (additional events on top of Base)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

**Category: Workflows**

Coverage level	Events logged
Base	Workflow scheme created, Workflow scheme copied, Workflow scheme deleted, Workflow scheme updated,, Workflow scheme added to project, Workflow scheme removed from project, Workflow created, Workflow copied, Workflow deleted, Workflow updated, Workflow draft published, Workflow renamed
Advanced (additional events on top of Base)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

**Category: Notifications**

Coverage level	Events logged
Base	Notification scheme created, Notification scheme deleted, Notification scheme copied, Notification scheme updated, Notification added, Notification deleted, Notification scheme added to project, Notification scheme removed to project,
Advanced (additional events on top of Base)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

**Category: Issues**

Coverage level	Events logged
Base	Issue security scheme added to project, Issue security scheme removed from project
Advanced (additional events on top of Base)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

**Category: Global administration**

Coverage level	Events logged
Base	Product license changed, Evaluation license requested
Advanced (additional events on top of Base)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

**Category: Issue operations**

Coverage level	Events logged
Base	Issue link created, Issue link deleted
Advanced (additional events on top of Base)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

**Category: DVCS**

Coverage level	Events logged
Base	DVCS organization added, DVCS organization removed, OAuth credentials changed for organization  Repository disabled, Repository enabled, Smart commits enabled for repository, Smart commits disabled for repository  Auto-sync of new repositories enabled for organization, Auto-sync of new repositories disabled for organization, List of repositories refreshed for organization, Soft sync of repository performed, Full sync of repository performed, Repository marked as deleted, Scheduled sync or organizations performed
Advanced (additional events on top of Base)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

## User management

### Category: Users and groups

#### JIRA SERVICE MANAGEMENT

Coverage level	Events logged
Base	User created, User deleted, User updated, User renamed, User credentials updated, User anonymization started, User anonymized, Group created, Group deleted, Group added to group, User added to group, Group removed from group, User removed from group, Application role group deleted, Application role configuration cleared, Application role set
Advanced (additional events on top of Base)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

### Category: Permissions

#### JIRA SERVICE MANAGEMENT

Coverage level	Events logged
Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	User(s) or Group(s) added as Default Members for Project Role, User(s) or Group(s) removed from Default Members for Project Role
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

## Permissions

The Permissions coverage area only has one category.

Coverage level	Events logged
Base	Permission scheme created, Permission scheme added to project, Permission scheme removed from project
Advanced (additional events on top of Base)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

## Local configuration and administration

### Category: Projects

Coverage level	Events logged
Base	Project created, Project deleted, Project updated, Project category assigned to project, Project avatar updated, Project archived, Project restored, Sprint deleted, Board created, Board deleted, Component created, Component deleted, Component updated, Component merged, Component archived, Component restored, Version archived, Version restored, Version created, Version deleted, Version updated, Version merged, Version released, Version unreleased
Advanced (additional events on top of Base)	Default assignee type updated, Project lead updated
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

### Category: Indexing

Coverage level	Events logged
Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	Project reindex started, Project reindex finished
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

### Category: SLAs

#### JIRA SERVICE MANAGEMENT

Coverage level	Events logged
Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	SLA calendar created, SLA calendar deleted, SLA calendar updated, SLA goal created, SLA goal deleted, SLA goal updated, SLA name updated, SLA conditions created, SLA conditions updated
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS LOGGED HERE

### Category: Reports

#### JIRA SERVICE MANAGEMENT

Coverage level	Events logged
Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	Service project report created, Service project report deleted, Service project report updated

Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS LOGGED HERE
--	--

**Category: Agents**

## JIRA SERVICE MANAGEMENT

Coverage level	Events logged
Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	Agent invited to the project, Agent removed from the project
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS LOGGED HERE

**Category: Email channels**

## JIRA SERVICE MANAGEMENT

Coverage level	Events logged
Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	Email channel enabled, Email channel disabled, Email channel updated (password changes)
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS LOGGED HERE

**Category: Request types**

## JIRA SERVICE MANAGEMENT

Coverage level	Events logged
Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	Request type created, Request type deleted
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS LOGGED HERE

**Category: Organizations**

## JIRA SERVICE MANAGEMENT

Coverage level	Events logged
Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	Organization created, Organization deleted, Organization updated, Organization membership changed, Organization associated with project, Organization disassociated from project
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS LOGGED HERE

**Category: Notification templates**

## JIRA SERVICE MANAGEMENT

Coverage level	Events logged
Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	JSM notification rule template updated
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS LOGGED HERE

## Security

## Category: Auditing

Coverage level	Events logged
Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	Audit log configuration updated, Audit log exported, Audit log search performed
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS LOGGED HERE

## Category: Login

Coverage level	Events logged
Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	Secure admin (websudo) login failed (8.12.0+)
Full (additional events on top of Base and Advanced)	User login successful, User logout, User login failed

## Category: Security

Coverage level	Events logged
Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	Secure admin access revoked, Secure admin access granted, OAuth 2.0 integration succeeded (8.12.0+), OAuth 2.0 integration failed (8.12.0+)
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS LOGGED HERE

## End user activity

## Category: Filters

Coverage level	Events logged
Base	Filter created, Filter updated, Filter deleted
Advanced (additional events on top of Base)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE
--	--

**Category: Dashboards**

Coverage level	Events logged
Base	Dashboard created, Dashboard updated, Dashboard deleted
Advanced (additional events on top of Base)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

**Category: Issues**

Coverage level	Events logged
Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	Issue deleted, Issue archived, Issue restored, Issue attachment deleted, Comment of another user deleted, Sub-task deleted, Issue link deleted, Archived issues exported (8.12.0+), Issues exported (8.12.0+)
Full (additional events on top of Base and Advanced)	Issue created, Issue comment added, Issue link created

**Category: Projects**

Coverage level	Events logged
Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	Own comment deleted
Full (additional events on top of Base and Advanced)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

**Category: Search**

Coverage level	Events logged
Base	NO EVENTS AVAILABLE
Advanced (additional events on top of Base)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE
Full (additional events on top of Base and Advanced)	

**Category: Data pipeline**

Coverage level	Events logged
Base	Full data export cancelled, Full data export triggered, Unauthorized full data export triggered, Full data export failed
Advanced (additional events on top of Base)	CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

Full (additional events on top of Base and Advanced)

CURRENTLY NO ADDITIONAL EVENTS AVAILABLE

# Audit log integrations in Jira

Jira Data Center writes audit logs to the database and a log file. By itself, the log file saves you the effort of periodically exporting your audit logs from the database for long-term storage. However, the main purpose of the file is to easily integrate Jira Data Center to a third-party logging platform.

## Event coverage and log retention

The Audit log settings menu controls the coverage of audit logs in both database and log file. However, this menu does not control the log file's retention period.

The log file's retention is ultimately controlled by [log rotation](#). We use basic log rotation to manage the volume of logs. We automatically archive the audit logfile when:

- the node's time reaches 12:00 midnight, or
- the audit log file reaches 100MB.

Once a node reaches the log file retention limit, the oldest one is deleted. By default the limit is 100 log files (the current audit log file + 99 archives). Make sure you allocate enough disk space for these log files on each application node. For the default setting of 100 files, you should allow 10GB.

## Log file details

Jira Data Center writes audit logs in real time to the home directory. Specifically, these logs are written to the audit logfile. On clustered Jira Data Center deployments, each application node will produce its own logfile in its local home directory.

### Location

To integrate the audit logfile with a third-party logging platform, you'll need to know its exact location. This may vary, depending on how you configured your home directory. For more information about the local home directory, [click here](#).

On a clustered Jira Data Center deployment, the audit logfile's directory should be the same on all nodes.

See [CloudWatch Logs Agent Reference](#) for more information. If you want to see how we automate this via Ansible, check out our deployment playbooks on <https://bitbucket.org/atlassian/dc-deployments-automation/src/master/>.

### File name

The audit log file name uses the following naming convention:

```
YYYYMMDD-XXXXX.audit.log
```

The `XXXXX` portion is a 5-digit number (starting with `00000`) tracking the number of audit log files archived in the same day (`YYYYMMDD`). For example, if there are 5 archived log files today (January 1, 2020), then:

- the oldest archived log file is `20200101.00000.audit.log`
- the current audit log file is `20200101.00005.audit.log`

### Format

Each audit log is written as a JSON entry to the audit log file. Every line in the file represents a single event, allowing you to use [regular expressions](#) to do simple searches if needed.

## Integrating with logging agents

Most enterprise environments use a third-party logging platform to aggregate, store, and otherwise manage logs from all hosts. Logging platforms like AWS CloudWatch and Splunk use *agents* to collect logs from every host in the environment. These agents are installed on each host, collecting local logs and sending them back to a centralized location to be aggregated, analyzed, audited, and/or stored.

If your logging platform uses agents this way, you can configure each node's agent to monitor the audit log file directly. Logging agents from most major platforms (including AWS CloudWatch, Splunk, ELK, and Sumo Logic) are compatible with the audit logfile.

### Amazon CloudWatch Agent

We provide [Quick Starts for Jira Data Center](#) for easy deployments on AWS. This Quick Start lets you deploy Jira Data Center along with an Amazon CloudWatch instance to monitor it.

To set up Amazon CloudWatch, use the **Enable CloudWatch Integration** parameter's default setting (namely, `Metrics and Logs`). The Quick Start will then configure the [Amazon CloudWatch Agent](#) to collect the logs from each node's audit log files. The agent will send these logs to a separate log group named `jira-software-<aws-stack-name>-audit`.

Our Quick Start also sets up a default dashboard to help you read the collected data, including logs from each audit logfile. Refer to [Working With Log Groups and Log Streams](#) for related information.

#### Manual configuration

If needed, you can also manually configure the Amazon CloudWatch agent to collect the audit log files. To do this, set the following parameters in the Agent Configuration File:

- `file`: set this to `<local home directory>/log/audit/*`. Don't forget to set the absolute path to the [home directory](#).
- `log_group_name` and `log_stream_name`: use these to send Jira Data Center's audit logs to a specific log group or stream.

### Splunk Universal Forwarder

For Splunk Enterprise or Splunk Cloud, you can use the [Splunk Universal Forwarder](#) as your logging agent. This will involve installing the universal forwarder on each application node.

You'll also need to define each node's audit log directory as one of the forwarder's inputs. This will set the forwarder to send all logs from the audit log directory to a pre-configured [receiver](#). One way to define the forwarder's inputs is through the Splunk CLI. For Linux systems, use the following command on each application node:

```
./splunk add monitor <local home directory>/log/audit/*audit.log
```

Refer to the following links for detailed instructions on configuring the Splunk Universal Forwarder on each node:

- [How to forward data to Splunk Enterprise](#)
- [How to forward data to Splunk Cloud](#)

### Filebeat (for the ELK stack)

Within the [ELK stack](#), you can use the [Filebeat](#) plugin to collect logs from each node's audit log files. Each time a log is written to the current audit log file, Filebeat will forward that log to Elasticsearch or Logstash.

To set this up, [install Filebeat](#) first on each application node. Then, set the audit log file directory as a [Filebeat input](#). To do that, add its directory as a path in the `filebeat.inputs` section of each node's `filebeat.yml` configuration file. For example:

```
filebeat.inputs:  
- type: log  
  enabled:true  
  paths:  
  - <local home directory>/log/audit/
```

### Sumo Logic installed collectors

If you have a Sumo Logic instance, you can use [installed collectors](#) to collect logs from each node's audit log files. To do this, [install a collector](#) on each node first. Then, add `<local home directory>/log/audit/*` as a [Local File Source](#) to each node's collector.

# Data pipeline

 This feature is available with a Jira Data Center license.

Data pipeline provides an easy way to export data from Jira, Confluence, or Bitbucket, and feed it into your existing data platform (like [Tableau](#) or [PowerBI](#)). This allows you to:

- generate richer reports and visualizations of site activity
- better understand how your teams are using your application
- make better decisions on optimizing the use of Jira or Confluence in your organization

You can trigger a data export in your applications admin console or through the REST API. Data will be exported in CSV format. You can only perform one data export at a time.

For a detailed reference of the exported data's schema, see [Data pipeline export schema](#).

Data pipeline is available in Data Center editions of:

- Jira 8.14 and later
- Confluence 7.12 and later
- Bitbucket 7.13 and later

## Requirements

To export data using the data pipeline, you'll need:

- A valid Jira Data Center license
- [Jira system administrator](#) permissions. See [Security overview](#) for more information about supported API authentication methods.

## Considerations

There are a number of security and performance impacts you'll need to consider before getting started.

### Security

The export will include all data, including PII (Personally Identifiable Information) and restricted content. This is to provide you with as much data as possible, so you can filter and transform to generate the insights you're after.

If you need to filter out data based on security and confidentiality, this must be done after the data is exported.

Exported files are saved in your shared home directory, so you'll also want to check this is secured appropriately.

### Export performance

Exporting data can take a long time in large instances. We intentionally export data at a limited rate to keep any performance impact to your site under a 5% threshold. It's important to note that there is no impact to performance unless an export is in progress.

When scheduling your exports, we recommend that you:

- Limit the amount of data exported using the `fromDate` parameter, as a date further in the past will export more data, resulting in a longer data export.

### On this page:

- [Requirements](#)
- [Considerations](#)
- [Access the data pipeline](#)
- [Schedule regular exports](#)
- [Check the status of an export](#)
- [Cancel an export](#)
- [Configuring the data export](#)
- [Use the data pipeline REST API](#)
- [Output files](#)
- [Analyse data pipeline data](#)
- [Troubleshooting issues with data exports](#)

- Schedule exports during hours of low activity, or on a node with no activity, if you do observe any performance degradation during the export.

Export scope	Approximate export duration	
	1 million issues	7 million issues
<b>Jira Software</b>		
<ul style="list-style-type: none"> <li>• Without custom fields</li> <li>• Without issue history</li> </ul>	15 minutes	2 hours
<ul style="list-style-type: none"> <li>• With custom fields</li> <li>• Without issue history</li> </ul>	1 hour	9 hours
<ul style="list-style-type: none"> <li>• With custom fields</li> <li>• With issue history</li> </ul>	5 hours	22 hours
<b>Jira Software and Jira Service Management</b>		
<ul style="list-style-type: none"> <li>• Without custom fields</li> <li>• Without issue history</li> </ul>	30 minutes to 2 hours	3 to 6 hours

### Test performance VS production

The data presented here is based on our own internal regression testing. The actual duration and impact of a data export on your own environment will likely differ depending on:

- your infrastructure, configuration, and load
- applications installed (Jira Software and Jira Service Management)
- amount of custom field and issue history data to be exported.

We used [Jira Performance Tests](#) to test a data export's performance on a Jira Data Center environment on AWS. This environment had one [c5.9xlarge](#) Jira node and one PostgreSQL database. To test user load, we used 24 virtual users across 2 virtual user nodes.

## Access the data pipeline

To access the data pipeline:

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. Select **Data pipeline**.

## Schedule regular exports

The way to get the most value out of the data pipeline is to schedule regular exports. The data pipeline performs a full export every time, so if you have a large site, you may want to only export once a week.

To set the export schedule:

1. From the Data pipeline screen, select **Schedule settings**.
2. Select the **Schedule regular exports** checkbox.
3. Select the date to include data from. Data from before this date won't be included. This is usually set to 12 months or less.

4. Choose how often to repeat the export.
5. Select a time to start the export. You may want to schedule the export to happen outside working hours.
6. Select the **Schema version** to use (if more than one schema is available).
7. **Save** your schedule.

### Schedule settings

Regular exports will help you see how your data is changing over time. Set up a schedule to automatically export data at a time that suits your organisation. [Learn more about exporting data](#)

**i** The export process can take several hours, and may impact performance. Consider scheduling your export for a time when Jira is less active.

Schedule regular exports

Repeat every \*

weeks

Repeat on

Sun  Mon  Tue  Wed  Thu  Fri  Sat

Repeat at \*

Include data from \*

Schema \*

## Timezones and recurring exports

We use your server timezone to schedule exports (or system timezone if you've overridden the server time in the application). The export schedule isn't updated if you change your timezone. If you do need to change the timezone, you'll need to edit the schedule and re-enter the export time.

You can schedule exports to happen as often as you need. If you choose to export on multiple days, the first export will occur on the nearest day after you save the schedule. Using the example in the screenshot above, if you set up your schedule on Thursday, the first export would occur on Saturday, and the second export on Monday. We don't wait for the start of the week.

## Export schema

The export schema defines the structure of the export. We version the schema so that you know your export will have the same structure as previous exports. This helps you avoid problems if you've built dashboards or reports based on this data.

We only introduce new schema versions for breaking changes, such as removing a field, or if the way the data is structured changes. New fields are simply added to the latest schema version.

Older schema versions will be marked as deprecated, and may be removed in future versions. You can still export using these versions, just be aware we won't update them with any new fields.

## Check the status of an export

You can check the status of an export and view when your last export ran from the data pipeline screen.

The **Export details** table will show the most recent exports, and the current status.

### Data pipeline Schedule settings

Use the data pipeline to export data from this site for analysis in your favourite business intelligence platform. You can export using the REST API or schedule regular exports. [Learn more about the data schema](#)

Exporting data may impact performance, so we recommend scheduling exports when Jira is less active. [Learn more about exporting data](#)

#### Schedule

Next export date **14 August 2021, 2:00 am AEST**  
 Repeat every **1 week**  
 Repeat on **Monday, Saturday**  
 Repeat at **2:00 am AEST**  
 Schema **Version 2**

#### Export details

Job ID	Start date	End date	Node	Schema version	Root export path	Job status	Actions
8	2 Aug 2021, 9:17:00 am AEST	2 Aug 2021, 9:17:00 am AEST	defaultNode	2	/Jira/localhome/data-pipeline/export	COMPLETED	...
7	31 July 2021, 12:30:59 pm AEST	31 July 2021, 12:30:59 pm AEST	defaultNode	2	/Jira/localhome/data-pipeline/export	COMPLETED	...
6	30 July 2021, 9:17:00 am AEST	30 July 2021, 9:17:00 am AEST	defaultNode	2	/Jira/localhome/data-pipeline/export	COMPLETED	...
5	29 July 2021, 4:26:00 pm AEST	29 July 2021, 4:26:00 pm AEST	defaultNode	2	/Jira/localhome/data-pipeline/export	COMPLETED	...
4	28 July 2021, 1:14:00 pm AEST	28 July 2021, 1:14:00 pm AEST	defaultNode	1	/Jira/localhome/data-pipeline/export	COMPLETED	...

Select **...** > **View details** to see the full details of the export in JSON format. Details include the export parameters, status, and any errors returned if the export failed.

For help resolving failed or cancelled exports, see [Data pipeline troubleshooting](#).

## Cancel an export

To cancel an export while it is in progress:

- Go to the **Data pipeline** screen.
- Select **...** next to the export, and choose **Cancel** export.
- Confirm you want to cancel the export.

It can take a few minutes for the processes to be terminated. Any files already written will remain in the export directory. You can delete these files if you dont need them.

## Automatic data export cancellations

If you shut down a node running a data export, the export will be cancelled. However, if the JVM is not notified after a crash or hardware-level failure, the export process may get locked. This means you'll need to manually mark the export as cancelled (through the UI, or via the REST API by making a `DELETE` request). This releases the process lock, allowing you to perform another data export.

## Configuring the dataexport

You can configure the format of the export data through the following system properties.

Default value	Description
	plugin.data.pipeline.embedded.line.break.preserve
false	Specifies whether embedded line breaks should be preserved in the output files. Line breaks can be problematic for some tools such as Hadoop.  This property is set to <code>False</code> by default, which means that line breaks are escaped.
	plugin.data.pipeline.embedded.line.break.escape.char

\\n	Escaping character for embedded line breaks. By default, we'll print\\nfor every embedded line break.
plugin.data.pipeline.minimum.usable.disk.space.after.export	
5GB	To prevent you from running out of disk space, the data pipeline will check before and during an export that there is at least 5GB free disk space.  Set this property, in gigabytes, to increase or decrease the limit. To disable this check, set this property to -1(not recommended).

You can further configure your export to exclude certain types of data using feature flags. See [How to manage dark features in Jira](#) to learn how to use feature flags.

Default value	Description
data.pipeline.feature.jira.all.exportable.custom.fields.enabled	
Enabled	Specifies whether custom field data should be included in the export. Exporting custom field data may increase your export duration, depending on the amount of custom field data you have.  Change the <code>.enabled</code> suffix to <code>.disabled</code> to exclude custom field data from your export.
data.pipeline.feature.jira.issue.history.export.enabled	
Enabled	Specifies whether historical issue data should be included in the export. Exporting historical data will significantly increase your export duration.  Change the <code>.enabled</code> suffix to <code>.disabled</code> to exclude issue history from your export.
data.pipeline.feature.jira.archived.issue.export.enabled	
Disabled	Specifies whether archived issues should be included in the export.  Add the flag with the suffix <code>.enabled</code> to include archived issues in your export.

## Use the data pipeline REST API

You can use the data pipeline REST API to export data.

To start a data pipeline export, make a POST request to `<base-url>/rest/datapipeline/latest/export`.

Here is an example request, using cURL and a personal access token for authentication:

```
curl -H "Authorization:Bearer ABCD1234" -H "X-Atlassian-Token: no-check"
-X POST https://myexamplesite.com/rest/datapipeline/latest/
export?fromDate=2020-10-22T01:30:11Z
```

You can also use the API to check the status, change the export location, and schedule or cancel an export.

For full details, refer to the [Data pipeline REST API reference](#).

## Output files

Each time you perform a data export, we assign a numerical job ID to the task (starting with 1 for your first ever data export). This job ID is used in the file name, and location of the files containing your exported data.

## Location of exported files

Exported data is saved as separate CSV files. The files are saved to the following directory:

- `<shared-home>/data-pipeline/export/<job-id>` if you run Jira in a cluster
- `<local-home>/data-pipeline/export/<job-id>` if you are using non-clustered Jira

Within the `<job-id>` directory you will see the following files:

- `issues_<job-id>_<schema_version>_<timestamp>.csv`
- `issue_fields_<job-id>_<schema_version>_<timestamp>.csv`
- `issue_history_<job-id>_<schema_version>_<timestamp>.csv`
- `issue_links_<job-id>_<schema_version>_<timestamp>.csv`
- `sla_cycles_<job-id>_<schema_version>_<timestamp>.csv` (Jira Service Management only)
- `users_<job-id>_<schema_version>_<timestamp>.csv`

To load and transform the data in this export, you'll need to understand its schema. See [Data pipeline export schema](#) for a summary of the contents of each file.

### Set a custom export path

By default, the data pipeline exports the files to the home directory, but you can use the REST API to set a custom export path.

To change the root export path, make a PUT request to `<base-url>/rest/datapipeline/1.0/config/export-path`.

In the body of the request pass the absolute path to your preferred directory.

For full details, including how to revert back to the default path, refer to the [Data pipeline REST API reference](#).

### Analyse data pipeline data

Once you've scheduled your exports, and have the CSV files, you can import these files into a database or data lake for analysis.

### Sample DevOps dashboards

To get you started, we've created a DevOps dashboard template for Tableau and Microsoft PowerBI that uses Jira data to give you an insight into the engineering health of your team.

[Make the most of the data pipeline with the DevOps dashboard](#)

### Sample Spark and Hadoop import configurations

If you have an existing Spark or Hadoop instance, use the following references to configure how to import your data for further transformation:

#### Sample Notebook Configuration

```
%python
# File location
file_location = "/FileStore/**/export_2020_09_24T03_32_18Z.csv"

# Automatically set data type for columns
infer_schema = "true"
# Skip first row as it's a header
first_row_is_header = "true"
# Ignore multiline within double quotes
multiline_support = "true"

# The applied options are for CSV files. For other file types, these will be ignored. Note escape &
# quote options for RFC-4801 compliant files
df = spark.read.format("csv") \
    .option("inferSchema", infer_schema) \
```

```

.option("header", first_row_is_header) \
.option("multiLine", multiline_support) \
.option("quote", "\"" ) \
.option("escape", "\"" ) \
.option("encoding", "UTF-8").load(file_location)

display(df)

```

### Create table script

```

CREATE EXTERNAL TABLE IF NOT EXISTS some_db.datapipeline_export (
  `id` string,
  `instance_url` string,
  `key` string,
  `url` string,
  `project_key` string,
  `project_name` string,
  `project_type` string,
  `project_category` string,
  `issue_type` string,
  `summary` string,
  `description` string,
  `environment` string,
  `creator_id` string,
  `creator_name` string,
  `reporter_id` string,
  `reporter_name` string,
  `assignee_id` string,
  `assignee_name` string,
  `status` string,
  `status_category` string,
  `priority_sequence` string,
  `priority_name` string,
  `resolution` string,
  `watcher_count` string,
  `vote_count` string,
  `created_date` string,
  `resolution_date` string,
  `updated_date` string,
  `due_date` string,
  `estimate` string,
  `original_estimate` string,
  `time_spent` string,
  `parent_id` string,
  `security_level` string,
  `labels` string,
  `components` string,
  `affected_versions` string,
  `fix_versions` string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
  "escapeChar" = "\\",
  'quoteChar' = '"',
  'separatorChar' = ','
) LOCATION 's3://my-data-pipeline-bucket/test-exports/'
TBLPROPERTIES ('has_encrypted_data'='false');

```

## Troubleshooting issues with data exports

Exports can fail for a number of reasons, for example if your search index isn't up to date. For guidance on common failures, and how to resolve them, see [Data pipeline troubleshooting](#) in our knowledge base.

# Data pipeline export schema

This page describes the structure and data schema of the Jira data export files.

To learn more about how to set up and configure your data pipeline, see [Data pipeline](#).

The output files are written in CSV format and are RFC4180 compliant. They have the following characteristics:

- Each file has a header. This includes files from exports that resulted in no data.
- New lines are separated by CRLF characters `\r\n`.
- Fields containing line breaks (CRLF), double quotes, and commas are enclosed in double quote.
- If double-quotes are present inside fields, then a double-quote appearing inside a field are escaped by preceding it with another double quote. For example: "aaa", "b" "bb", "ccc".
- Fields with no data (null values) are represented in the CSV export by two consecutive delimiters (as in, , , ).
- Embedded break lines are escaped by default and printed as n.

## Availability

The data exported depends on your Jira application and version.

Table	Jira Software	Jira Service Management
Issues	8.14	4.15
Fields		
• system fields	8.14	4.15
• custom fields	8.17	4.17
Issue history	8.18	4.18
Issue links	8.19	4.19
SLAs	N/A	4.15
Users	8.19 (schema 2)	4.19 (schema 2)

Individual fields are available in all schema versions, unless specifically noted in the tables below.

## Issues schema

All data relating to issues will be exported to `issues` CSV file.

Field	Description
-------	-------------

### On this page:

- [Issues schema](#)
- [Fields schema for Jira Software and Jira Service Management fields](#)
- [Issue history schema](#)
- [Issue links](#)
- [SLA Cycle data schema \(for Jira Service Management only\)](#)
- [Users schema](#)

id ( <b>Primary Key</b> )	<p><b>Type:</b> Number</p> <p><b>Description:</b> Unique ID of this issue. Use as <b>Primary Key</b>.</p> <p><b>Example:</b> 10002</p>
instance_url	<p><b>Type:</b> String</p> <p><b>Description:</b> Base url of the current instance</p> <p><b>Example:</b> http://www.example.com/jira</p>
url	<p><b>Type:</b> String</p> <p><b>Description:</b> URL of the issue</p> <p><b>Example:</b> http://www.example.com/jira/browse/TEST-1002</p>
key	<p><b>Type:</b> String</p> <p><b>Description:</b> Unique Key for this issue</p> <p><b>Example:</b> TEST-1002</p>
project_key	<p><b>Type:</b> String</p> <p><b>Description:</b> Key of the project this issue is in</p> <p><b>Example:</b> TEST</p>
project_name	<p><b>Type:</b> String</p> <p><b>Description:</b> Title of the project this issue is in</p> <p><b>Example:</b> Sample Test Project</p>
project_type	<p><b>Type:</b> String</p> <p><b>Description:</b> Type of the project this issue is in (can be Business, Software, Service Desk)</p> <p><b>Example:</b> Software</p>
project_category	<p><b>Type:</b> String</p> <p><b>Description:</b> Assigned category for the project this issue is in</p> <p><b>Example:</b> Sample Project Category</p>
issue_type	<p><b>Type:</b> String</p> <p><b>Description:</b> Type of issue (for example, task, bug, or epic)</p> <p><b>Example:</b> Task</p>
summary	<p><b>Type:</b> String</p> <p><b>Description:</b> Summary of the issue</p> <p><b>Example:</b> Sample issue</p>

description	<p><b>Type:</b> String</p> <p><b>Description:</b> Description of the issue (limited to 2000 characters)</p> <p><b>Example:</b> This is a sample issue for demo purposes.</p>
creator_id	<p><b>Type:</b> Number</p> <p><b>Description:</b> Unique identifier of the user who created the issue, regardless of directory</p> <p><b>Example:</b> 10000</p>
creator_name	<p><b>Type:</b> String</p> <p><b>Description:</b> Name of the user issue creator</p> <p><b>Example:</b> Alana Grant</p> <p><b>Schema:</b> Version 1 only. This data is now contained in the Users file.</p>
reporter_id	<p><b>Type:</b> Number</p> <p><b>Description:</b> Unique identifier of the issue reporter, regardless of directory</p> <p><b>Example:</b> 10001</p>
reporter_name	<p><b>Type:</b> String</p> <p><b>Description:</b> Name of the issue reporter</p> <p><b>Example:</b> Omar Darboe</p> <p><b>Schema:</b> Version 1 only. This data is now contained in the Users file.</p>
assignee_id	<p><b>Type:</b> Number</p> <p><b>Description:</b> Unique identifier of the issue assignee, regardless of directory</p> <p><b>Example:</b> 10002</p>
assignee_name	<p><b>Type:</b> String</p> <p><b>Description:</b> Name of the issue assignee</p> <p><b>Example:</b> Jie Yang Song</p> <p><b>Schema:</b> Version 1 only. This data is now contained in the Users file.</p>
status	<p><b>Type:</b> String</p> <p><b>Description:</b> Current status of this issue</p> <p><b>Example:</b> Not Started</p>
status_category	<p><b>Type:</b> String</p> <p><b>Description:</b> Status category for the current status for this issue</p> <p><b>Example:</b> In Progress</p>

priority_sequence	<p><b>Type:</b> String</p> <p><b>Description:</b> The priority order (admins can change this which will result in non-deterministic priority orders)</p> <p><b>Example:</b> 3</p>
priority_name	<p><b>Type:</b> String</p> <p><b>Description:</b> Name of the priority for this issue</p> <p><b>Example:</b> Medium</p>
resolution	<p><b>Type:</b> String</p> <p><b>Description:</b> The final status of an issue. Jira considers the workflow of an Issue complete once a value is present in the resolution field.</p> <p><b>Example:</b> Done</p>
watcher_count	<p><b>Type:</b> Number</p> <p><b>Description:</b> Number of users watching this issue</p> <p><b>Example:</b> 5</p>
vote_count	<p><b>Type:</b> Number</p> <p><b>Description:</b> Number of votes for this issue</p> <p><b>Example:</b> 4</p>
created_date	<p><b>Type:</b> Date</p> <p><b>Description:</b> UTC timezone creation date of this issue (ISO string)</p> <p><b>Example:</b> 2020-11-05T14:44:57Z</p>
resolution_date	<p><b>Type:</b> Date</p> <p><b>Description:</b> UTC timezone resolution date of this issue (ISO string)</p> <p><b>Example:</b> 2020-11-05T14:44:57Z</p>
updated_date	<p><b>Type:</b> Date</p> <p><b>Description:</b> UTC timezone date that this issue was last updated (ISO string)</p> <p><b>Example:</b> 2020-11-05T14:44:57Z</p>
due_date	<p><b>Type:</b> Date</p> <p><b>Description:</b> UTC timezone date that this issue is due to be completed (ISO string)</p> <p><b>Example:</b> 2020-11-05T14:44:57Z</p>
estimate	<p><b>Type:</b> Number</p> <p><b>Description:</b> Estimate time remaining from original estimate, in seconds (requires time-tracking enabled)</p> <p><b>Example:</b> 5000</p>

original_estimate	<p><b>Type:</b> Number</p> <p><b>Description:</b> Original estimate that was set, in seconds (requires time-tracking enabled)</p> <p><b>Example:</b> 2000</p>
time_spent	<p><b>Type:</b> Number</p> <p><b>Description:</b> Amount of logged work, in seconds (requires time-tracking enabled). Field is null if there is no time spent.</p> <p><b>Example:</b> 4000</p>
labels	<p><b>Type:</b> String</p> <p><b>Description:</b> JSON array of label names. Field is null if there are no labels.</p> <p><b>Example:</b> ["test-label", "test2-label"]</p>
components	<p><b>Type:</b> String</p> <p><b>Description:</b> JSON array of component names. Field is null if there are no labels.</p> <p><b>Example:</b> ["component_a", "component_b"]</p>
parent_id	<p><b>Type:</b> Number</p> <p><b>Description:</b> ID of the parent issue. Field is null if the issue is not a subtask.</p> <p><b>Example:</b> 10001</p>
environment	<p><b>Type:</b> String</p> <p><b>Description:</b> A short description of the environment in which the issue occurred (for example, IE9 on Windows 7). This field has a limit of 2000 characters</p> <p><b>Example:</b> "linux"</p>
affected_versions	<p><b>Type:</b> String</p> <p><b>Description:</b> JSON Array of Affected Version Names. Field is null if there are no labels.</p> <p><b>Example:</b> ["a_version_1", "a_version_2"]</p>
fix_versions	<p><b>Type:</b> String</p> <p><b>Description:</b> JSON array of fix version names. Field is null if there are no labels.</p> <p><b>Example:</b> ["f_version_1", "f_version_2"]</p>
security_level	<p><b>Type:</b> String</p> <p><b>Description:</b> Security level name. <b>NOTE:</b> This can change between data exports.</p> <p><b>Example:</b> staff-only</p>
archived_by_user_id	<p><b>Type:</b> Number</p> <p><b>Description:</b> Unique identifier of the user who archived the issue, regardless of directory.</p> <p><b>Example:</b> 10002</p> <p><b>Schema:</b> Version 1 and later (requires Jira 8.19 or later)</p>

archived_date	<p><b>Type:</b> Date</p> <p><b>Description:</b> UTC timezone date this issue was archived (ISO string).</p> <p><b>Example:</b>2020-11-05T14:44:57Z</p> <p><b>Schema:</b> Version 1 and later(requires Jira 8.19 or later)</p>
---------------	---

## Fields schema for Jira Software and Jira Service Management fields

Data from Jira Software and Jira Service Management fields will be exported to the `issue_fieldsCSV` file and every custom field is displayed on a separate row. To map a custom field to its corresponding issue, use the `issue_id` from this file and find its matching `ID` from the Issues file.

User-generated custom fields, and fields provided by apps are generally included, as long as they are an exportable type (fields that implement `ExportableCustomFieldsType`).

If a custom field contains no data, it will not be exported.

Field	Description
issue_id	<p><b>Type:</b> Number</p> <p><b>Description:</b> ID to link to the corresponding issue in Issues data schema.</p> <p><b>Example:</b> 10002</p>
field_id	<p><b>Type:</b> String</p> <p><b>Description:</b> Pre-defined IDs of the Jira Software and Jira Service Management fields.</p> <p><b>Example:</b> <code>story_points</code></p>
field_name	<p><b>Type:</b> String</p> <p><b>Description:</b> Name of the field</p> <p><b>Example:</b> <code>Story Points</code></p>
field_value	<p><b>Type:</b> Number, String, JSON array</p> <p><b>Description:</b> Value of the field</p> <p><b>Examples:</b> <code>4 / [{"sprintName": "A"}, {"sprintName": "B"}]/5000</code></p>
The following Jira Software fields relate to non-epic issues.	

sprint	<p><b>Type:</b> JSON</p> <p><b>Description:</b> Sprints that the issue is currently in or has been in. All date and times are formatted as an ISO string with UTC timezone.</p> <p><b>Example:</b></p> <pre>[   {     "id": 2,     "name": "Sample Sprint 1",     "goal": "",     "boardId": 1,     "state": "FUTURE",     "startDate": "2020-09-20T19:44:13Z",     "endDate": "2020-10-04T18:44:13Z",     "endDate": "2020-10-04T18:44:13Z"   } ]</pre>
epic_link_id	<p><b>Type:</b> Number</p> <p><b>Description:</b> ID of the epic linked to this issue</p> <p><b>Example:</b> 10000</p>
story_points	<p><b>Type:</b> Number</p> <p><b>Description:</b> Number of story points associated with this issue</p> <p><b>Example:</b> 5</p>
The following Jira Software fields relate to epics	
epic_color	<p><b>Type:</b> String</p> <p><b>Description:</b> Color of the epic</p> <p><b>Example:</b> ghx-label-3</p>
epic_name	<p><b>Type:</b> String</p> <p><b>Description:</b> Name of the epic</p> <p><b>Example:</b> Test Epic</p>
epic_status	<p><b>Type:</b> String</p> <p><b>Description:</b> Status of the epic</p> <p><b>Example:</b> To Do</p>
The following fields relate to Jira Service Management	
customer_request_type	<p><b>Type:</b> String</p> <p><b>Description:</b> Customer Request Type for a JSM Customer Request</p> <p><b>Example:</b> Request new software</p>

organizations	<p><b>Type:</b> String</p> <p><b>Description:</b> List of organizations assigned to a given JSM Issue. Organizations are entities that admins can organize customers into.</p> <p><b>Example:</b> ["org1", "org2"]</p>
request_participants	<p><b>Type:</b> Number</p> <p><b>Description:</b> List of users IDs assigned as request participants for a given JSM Issue.</p> <p><b>Example:</b></p> <pre>[10100,10101,10000]</pre> <p><b>Schema:</b> Version 2 and later</p> <p>In schema version 1, this field contained the user ID and name. From schema version 2, name data is contained in the Users file.</p> <pre>[   {     "id": 10100,     "name": "User 1 display name"   },   {     "id": 10101,     "name": "user 2 display name"   } ]</pre>
satisfaction_comment	<p><b>Type:</b> String</p> <p><b>Description:</b> Comment given as part of a JSM Issue Satisfaction survey</p> <p><b>Example:</b> I got a prompt response. Thank you!</p>
satisfaction_rating	<p><b>Type:</b> String</p> <p><b>Description:</b> Rating given as part of a JSM Issue Satisfaction survey</p> <p><b>Example:</b> 4.0</p>
satisfaction_scale	<p><b>Type:</b> String</p> <p><b>Description:</b> Scale (max rating) the given satisfaction rating is based upon</p> <p><b>Example:</b> 5.0</p>
satisfaction_date	<p><b>Type:</b> Date</p> <p><b>Description:</b> Date the Satisfaction survey was completed</p> <p><b>Example:</b> 2020-10-04T18:44:13Z</p>

## Issue history schema

The history of each issue will be exported to `issue_history` CSV file. Only history after the export `fromDate` will be included.

Use the `issue_id` field to join this table to the `Issues` and `Fields` tables.

Field	Description
issue_id (foreign key)	<p><b>Type:</b> Number</p> <p><b>Description:</b> Unique ID of this issue. Link to the corresponding issue in the issues table.</p> <p><b>Example:</b> 10002</p>
changelog_id	<p><b>Type:</b> Number</p> <p><b>Description:</b> Identifier for the change, or group of changes that were made at the same time. Combine this with field to identify a single change.</p> <p><b>Example:</b> 10456</p>
author_id	<p><b>Type:</b> Number</p> <p><b>Description:</b> Unique identifier for the user who made the change.</p> <p><b>Example:</b> 10000</p>
author_key	<p><b>Type:</b> String</p> <p><b>Description:</b> Unique identifier of the author of the change as a unique string.</p> <p><b>Example:</b> JIRAUSER10000</p>
created_date	<p><b>Type:</b> Date</p> <p><b>Description:</b> UTC timezone date of this change (ISO string), truncated to minutes.</p> <p><b>Example:</b> 2021-05-12T23:01:42Z</p>
field	<p><b>Type:</b> String</p> <p><b>Description:</b> The name of the field that was changed.</p> <p><b>Example:</b> assignee</p>
field_type	<p><b>Type:</b> Number</p> <p><b>Description:</b> The type of field that was changed, either jira or custom</p> <p><b>Example:</b> jira</p>
from	<p><b>Type:</b> String</p> <p><b>Description:</b> Identifier for the value of the field before the change. Limited to 2000 characters.</p> <p><b>Example:</b> JIRAUSER10001</p>
from_string	<p><b>Type:</b> String</p> <p><b>Description:</b> Value of the field before the change, as a string. Limited to 2000 characters.</p> <p><b>Example:</b> Alana Grant</p>

to	<p><b>Type:</b> String</p> <p><b>Description:</b> Identifier for the value of the field after the change. Limited to 2000 characters.</p> <p><b>Example:</b> JIRAUSER10002</p>
to_string	<p><b>Type:</b> String</p> <p><b>Description:</b> Value of the field after the change, as a string. Limited to 2000 characters.</p> <p><b>Example:</b> Omar Darboe</p>
additional_information	<p><b>Type:</b> String</p> <p><b>Description:</b> Any additional information associated with the change, in JSON format.</p> <p><b>Example:</b></p> <pre>{   "old_status_category_id": "2",   "old_status_category_name": "New",   "new_status_category_id": "4",   "new_status_category_name": "In Progress" }</pre>

Note that exporting issue history can take some time, and there's a small chance that an issue may be updated after the details for that issue have been fetched. This can lead to the updated date in the issues file being earlier than the last change in the issue history file.

For example, DEV-456 is updated at 10:02:00, and again fifteen seconds later. If the data pipeline was fetching details of that issue at exactly that moment, the updated time in the issues file would be 10:02:00, but the issue history file would contain the subsequent change from 10:02:15.

## Issue links

All data relating to connections between issues will be exported to `issue_links` CSV file.

Field	Description
inward_issue_id	<p><b>Type:</b> Number</p> <p><b>Description:</b> Unique ID of the source issue.</p> <p><b>Example:</b> 10022</p>
outward_issue_id	<p><b>Type:</b> Number</p> <p><b>Description:</b> Unique ID of the destination issue.</p> <p><b>Example:</b> 10016</p>
issue_link_type_id	<p><b>Type:</b> Number</p> <p><b>Description:</b> Unique ID of the issue link type.</p> <p><b>Example:</b> 10000</p>
issue_link_type_name	<p><b>Type:</b> Number</p> <p><b>Description:</b> Issue link type, such as blocks, duplicates, relates to.</p> <p><b>Example:</b> blocks</p>

## SLA Cycle data schema (for Jira Service Management only)

Data relating to SLA Cycle fields from Jira Service Management will be exported to `sla_cyclesCSV` file. Each issue can have multiple SLAs, where each SLA cycle (`Ongoing` and/or `Completed`) is displayed on a separate row. To map an SLA to its corresponding issue, use the `issue_id` from this file and find its matching `ID` from the Issues file.

Field	Description
<code>issue_id</code>	<p><b>Type:</b> Number</p> <p><b>Description:</b> ID to link to the corresponding issue in Issues data schema</p> <p><b>Example:</b> 10002</p>
<code>sla_id</code>	<p><b>Type:</b> Number</p> <p><b>Description:</b> ID of the SLA of which the SLA cycle belongs to</p> <p><b>Example:</b> 15364</p>
<code>sla_name</code>	<p><b>Type:</b> String</p> <p><b>Description:</b> Name of the SLA of which the cycle belongs to.</p> <p><b>Example:</b> Time to first response</p>
<code>cycle_type</code>	<p><b>Type:</b> String</p> <p><b>Description:</b> Whether current SLA cycle is <code>Ongoing</code> or <code>Completed</code></p> <p><b>Example:</b> <code>Ongoing</code></p>
<code>start_time</code>	<p><b>Type:</b> Date</p> <p><b>Description:</b> Timestamp of when the SLA cycle started</p> <p><b>Example:</b> 2020-01-10T12:50:30Z</p>
<code>stop_time</code>	<p><b>Type:</b> Date</p> <p><b>Description:</b> Timestamp of when the SLA cycle transitioned from <code>Ongoing</code> to <code>Completed</code>. Only available for <code>Ongoing</code> cycles.</p> <p><b>Example:</b> 2020-01-10T12:50:30Z</p>
<code>paused</code>	<p><b>Type:</b> Boolean</p> <p><b>Description:</b> Notes whether the SLA cycle is paused or not. Can be <code>true</code>, <code>false</code>, or empty. Only available for <code>Ongoing</code> cycles.</p> <p><b>Example:</b> <code>True</code></p>
<code>remaining_time</code>	<p><b>Type:</b> Number</p> <p><b>Description:</b> Represents the time (in milliseconds) remaining before the expected SLA limit is breached. Remaining times are calculated and updated every 30 minutes. Therefore, the outputted value may not represent the actual current remaining time.</p> <p><b>Example:</b> 14400000</p>

elapsed_time	<p><b>Type:</b> Number</p> <p><b>Description:</b> Represents the time (in milliseconds) that has passed since the SLA cycle started.</p> <p><b>Example:</b> 14400000</p>
goal_duration	<p><b>Type:</b> Number</p> <p><b>Description:</b> Represents time (in milliseconds) taken to complete the current cycle. Only available for Completed cycles.</p> <p><b>Example:</b> 14400000</p>

## Users schema

All data relating to issues will be exported to usersCSV file.

Field	Description
user_id	<p><b>Type:</b> String</p> <p><b>Description:</b> ID of the user</p> <p><b>Example:</b> 10001</p> <p><b>Schema:</b> Version 2 and later</p>
instance_url	<p><b>Type:</b> URL</p> <p><b>Description:</b> Base URL of the current instance.</p> <p><b>Example:</b> https://yoursitename.com</p> <p><b>Schema:</b> Version 2 and later</p>
user_name	<p><b>Type:</b> String</p> <p><b>Description:</b> User name of the user.</p> <p><b>Example:</b> jsmith</p> <p><b>Schema:</b> Version 2 and later</p>
user_fullname	<p><b>Type:</b> String</p> <p><b>Description:</b> Full name of the user.</p> <p><b>Example:</b> John Smith</p> <p><b>Schema:</b> Version 2 and later</p>
user_email	<p><b>Type:</b> Email</p> <p><b>Description:</b> Email address of the user</p> <p><b>Example:</b> jsmith@example.com</p> <p><b>Schema:</b> Version 2 and later</p>

Disabled user accounts are not included in the export.

# Make the most of the data pipeline with the DevOps dashboard

The DevOps dashboard template gives you a glimpse of what's possible with the data pipeline. The template offers useful insights into the health of your engineering teams. We hope it will also provide a great jumping off point for creating your own dashboards and reports.

Download the template:

- [DevOps dashboard for Tableau](#)
- [DevOps Dashboard for PowerBI](#)

## On this page:

- [DevOps dashboard at a glance](#)
- [DevOps metrics in detail](#)
- [Predictability](#)
- [Configure the dashboard settings](#)

With the comprehensive data available for people, projects, and issues in the data pipeline, managers and business leaders can identify trends in productivity, quality, responsiveness, and predictability of work. While the Jira UI is focused on getting work done, dashboards and reports like this one provide insights on a macro level.

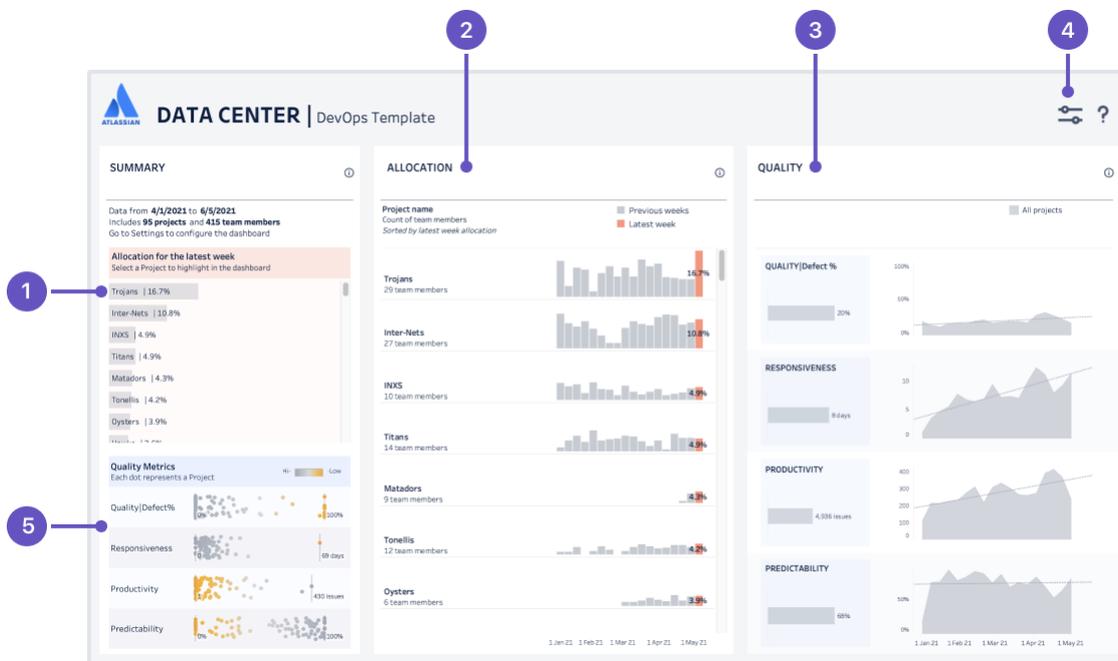
In this guide, we'll provide some explanation of the metrics we've included, how they're calculated, and what they might indicate about your team.

If you want to dive straight into connecting the template to your own data source, see:

- [Deploy the DevOps dashboard in Tableau](#)
- [Deploy the DevOps dashboard in PowerBI](#)

## DevOps dashboard at a glance

Here's the DevOps dashboard in Tableau, populated with the sample data included with the template.



1. **Project summary** lists the projects (or project categories if you choose to aggregate by category) featured in the dashboard, and shows the percentage of resources directed towards a project (measured by issues closed by team members). This list also acts as a filter for the report, if you want to drill down and see data for a specific project or project category.
2. **Allocation metrics** shows the percentage of resources directed towards each project (measured by issues closed by team members) to projects or project categories each week during the reporting period.

3. **Quality metrics** charts the quality, responsiveness, productivity and predictability metrics for all projects or project categories.
4. **Dashboard settings** configure the data to be included in the dashboard, including the date range, project or project category aggregation, and issue types to include.
5. **Quality summary** shows aggregate values for each of the four quality metrics by project, to help you identify outliers or systemic problems (or successes).

## DevOps metrics in detail

For the explanations below we'll assume you aggregate data in the dashboard by project. However, you can also choose to aggregate by project category. The concepts are the same for either option.

### Allocation

In the DevOps dashboard, the project allocation section shows the percentage of your team directed towards a project. This data can help you ensure you have coverage of your most important projects.

#### How is it calculated?

Because Data Center applications don't have the concept of a "team", we've used issues closed by a person as a proxy. We treat the people who closed issues in a project as team members for the purposes of that project. You can filter the report to only include particular people in a team or department.

It's worth noting that allocation is not indicating effort or time spent. The best way to explain how allocation is measured, is with a few examples.

First let's look at how allocation works for just one person. Jie closed a total of 10 issues across two projects during the time period.

User	Total issues closed	Project key	Issues closed in project	Allocation
Jie	10	InterNets	1	0.1
		Project Trojan	9	0.9

The allocation of this team of one therefore is 10% to the InterNets project, and 90% to Project Trojan.

Now let's add some more people and projects. In this example, we have a team of four people, who closed issues in four projects during the time period.

User	Total issues closed	Project key	Issues closed in project	Allocation
Alana	5	Oyster Project	5	1
Fran	7	Oyster Project	7	1
Jie	10	InterNets	1	0.1
		Project Trojan	9	0.9
Omar	2	InterNets	1	0.5
		Matadors	1	0.5

We get the project allocation by dividing the team members allocation by the total allocation, which in this example is 4 (1+1+0.1+0.9+0.5+0.5).

Project	Allocation
Oyster Project	50% (2/4)
InterNets	15% (0.6/4)

Project Trojan	23% (0.9/4)
Matador	13% (0.5/4)

This shows that 50% of your resources were directed towards the Oyster Project, and 15% to the InterNets project.

If you were to filter out the Oyster Project, the calculations change as follows.

User	Total issues closed	Project key	Issues closed in project	Allocation
Alana	-	-	-	-
Fran	-	-	-	-
Jie	10	InterNets	1	0.1
		Oyster Project	9	0.9
Omar	2	InterNets	1	0.5
		IOS	1	0.5

Now, 30% of resources (in the report scope) are directed towards the InterNets project.

Project	Allocation
InterNets	30% (0.6/2)
Project Trojan	45% (0.9/2)
Matadors	25% (0.5/2)

These calculations are all based on the person who completed an issue. Usually this is the assignee, but if a completed issue doesn't have an assignee, we'll use the issue history to determine who completed the issue. If there are multiple users in the issue history, we use the user who first completed the issue.

**What does it indicate?**

This metric helps you identify trends, and see where you may be able to re-deploy people onto projects with higher importance to the business.

This data can also be used in conjunction with the rest of the dashboard data, to see the impact of higher or lower allocation levels on other metrics.



Screenshot showing allocation metrics for two projects

### Quality (defects)

In the DevOps dashboard, we approach quality from an engineering health point of view. When left unchecked, technical debt will impact your teams ability to deliver quality software. By examining how you prioritize fixing defects over new feature work, you can spot trends that may indicate an engineering health problem.

#### How is it calculated?

 Quality = Defects resolved / total issues resolved x 100

Put simply, of all the issues resolved in a given week, what proportion were bugs and other defect issue types. Remember, were focused on engineering health indicators, not product or end-user experience quality.

You can define which issue types to treat as defects in the dashboard settings. For example, you may have different issue types for bugs, architectural tech debt, and failing tests that can all be classed as defects in the dashboard.

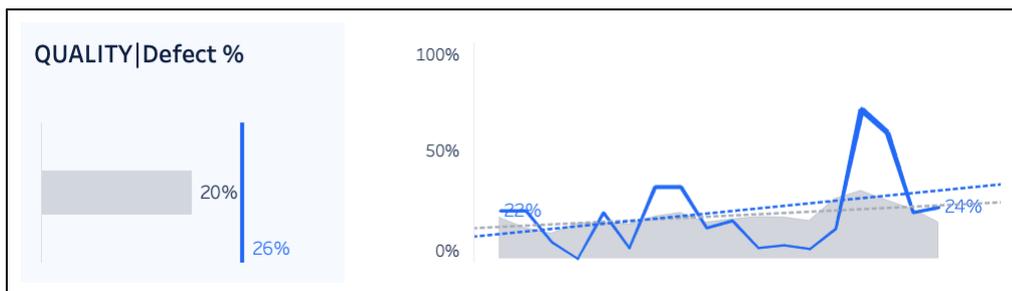
#### What does it indicate?

A steady line indicates the team is balancing technical debt with new feature work. Thats an indicator of a stable system with good engineering health.

If the line starts trending up (as in the example below), it might indicate that developers delaying working on defects until the end of a project, or you the teams focus has shifted to new feature work.

If the line starts trending down, developers may not be paying enough attention bugs, and you may have a quality issue in future.

If there are many sharp peaks and troughs, it would be worth delving deeper, and finding out what was happening during those weeks. There may be some team specific context youre missing, such as an incident or release deadline. You may also want to filter the dashboard by a specific project to see if the trend is noticeably different to the aggregate for all projects.



Screenshot showing Quality (defect) chart for one project, compared to all projects

### Responsiveness

In the DevOps dashboard, the responsiveness chart indicates how fast a team is completing work, by measuring the time to completion.

Responsive teams are extremely valuable to the business. They quickly deliver user facing value on a known cadence. With smart investments, you can build a strong and responsive team which enjoys the work, keeping customers and the business happy, without sacrificing the other dimensions long-term.

#### How is it calculated?

**i** Responsiveness = resolution date - created date

The value is expressed in days. Only issues that were completed within the given time frame are included.

#### What does it indicate?

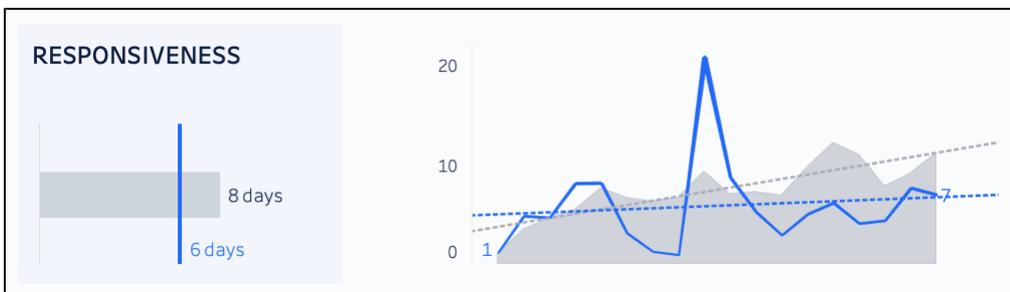
The solid lines indicate actual time to completion, and the dotted line indicates the direction its trending.

A steady line indicates the team is able to deliver customer facing value quickly. It may indicate good estimation practices, as work is broken down into similar sized, manageable chunks.

If the line is trending up, the team is taking longer to complete issues and deliver value. This may indicate an issue sizing and estimation problem in the team.

If the line is trending down, the team is taking less time to complete issues. Generally this is a good trend, but again, it can indicate an issue sizing or estimation problem.

Sharp peaks indicate the average time to complete an issue was higher than usual. It might be worth investigating what issues were completed in the time period, to see if the problem was caused by a single issue, or resourcing issues.



Screenshot showing the responsiveness chart for one project, compared to all projects.

## Productivity

In the DevOps dashboard, the productivity chart indicates the quantity of work being delivered.

#### How is it calculated?

**i** Productivity = number of issues completed

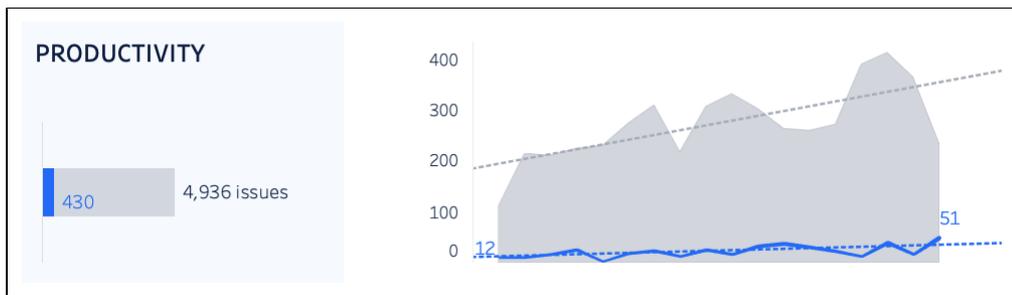
This is a simple metric that shows the volume work that meets the teams definition of done in the given time period.

#### What does it indicate?

A steady line indicates the team is working through a similar number of issues in a given time period. This can be an indicator of a stable workload, as a result of good estimation and sprint planning.

Sharp peaks and troughs may indicate a problem with workload, or with estimation. It could be worth delving deeper, and finding out what was happening during those weeks, and perhaps comparing this data to the allocation data for the same period.

Comparing a single project to all projects can be daunting, but focus on the shape of the line, rather than the difference in the number of issues.



Screenshot showing productivity chart for one project, compared to all projects

## Predictability

In the DevOps dashboard, the predictability chart indicates how consistent the pace of work is, by comparing the number of issues started (transitioned to an in progress status) with issues completed over time.

### How is it calculated?

**i** Predictability = issues completed - issues started

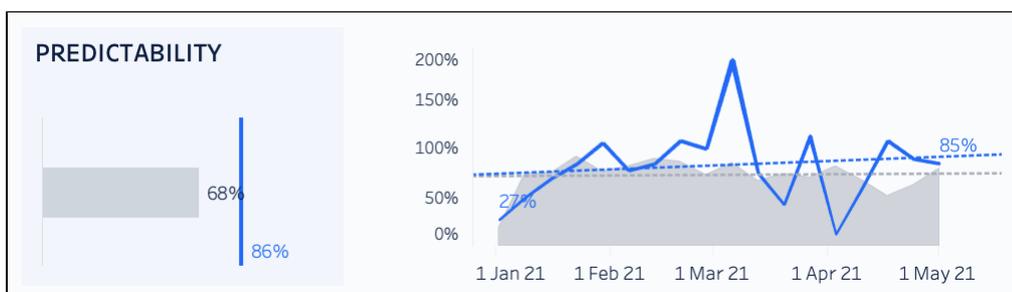
While responsiveness is measured in time (days), predictability focuses on the flow of work, specifically the volume of work that is started versus work completed within the same time frame.

### What does it indicate?

Ideally, the number of issues started and completed in a given period should be about the same, around 100% on the chart.

A value greater than 100% positive value indicates that more issues were completed than started. This is a good trend, but may indicate that people aren't picking up new tasks after completing a task.

A value lower than 100% indicates more issues were started than completed. This may indicate that work items are too large to be completed within the time frame, or that the team is starting too many things, and could benefit from some work in progress limits.



Screenshot showing predictability for a project, compared to all projects.

## Quality summary

In addition to the individual charts for quality, responsiveness, productivity, and predictability that allow you to compare one project to all projects, you can also see a summary view which shows the aggregate value for each project individually.

### What does it display?

**i** Aggregate for each project

For example, the responsiveness chart will show the aggregate time to completion for an individual project in the reporting timeframe.

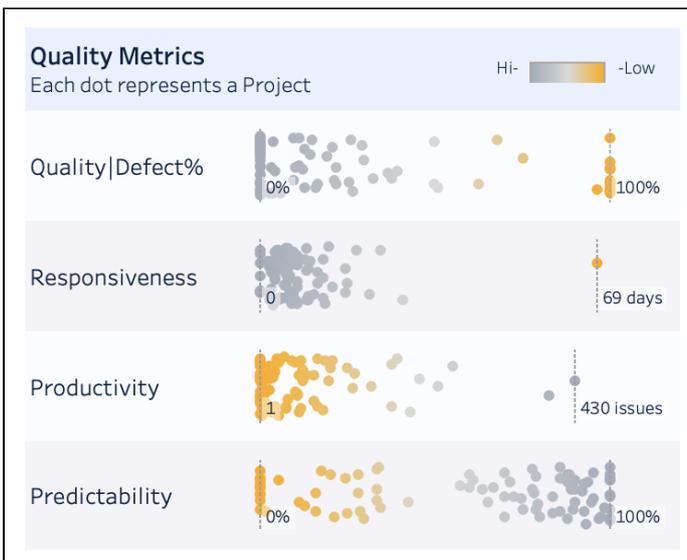
When reading this chart, remember that only the x-axis is important. The y-axis is spacing the projects out randomly, so the dots dont overlap. Hover over each dot to see the project name and value.

**What does it indicate?**

This helps you identify projects that are outliers, where it might be beneficial to investigate further.

You may also observe organization-wide trends that may indicate a bigger engineering health or agile craft problem.

As with the other charts, it can be useful to contrast the data with the allocation data. Perhaps the reason for low productivity and responsiveness is due to too few team members contributing to the project.



Screenshot showing the quality summary for all projects in the dashboard scope

**Configure the dashboard settings**

Once connected, you can configure the dashboard to show particular projects, team members, and time periods.

To configure the dashboard:

1. Select the **Settings** icon on the dashboard.
2. Set any date, project, team, and issue settings. Refer to the table below for information on each setting.
3. Select the **Close** icon to close the settings pane.

Here's a summary of the available settings.

Settings	Description
Date range	Drag the slider to set the start and end dates to report on. The dashboard can only show data for the date range included in the data pipeline export.
Time granularity	Select whether you want to display charts by week, month, quarter or year.
Dashboard aggregation	Select whether to report by project, or by project category.

Projects / project categories	If aggregating by project, select specific projects to report on. If aggregating by project category, select specific project categories to report on.  The list will include all projects or project categories included in the data pipeline export.
Team members	Select a sub-set of users to report on. By default this field returns user ID. You may be able to change the query to select by user name or full name, if that data is available.
Issue types	Select the issue types to include in the dashboard. For example you could choose to exclude Epics or Service requests.
Defect issue types	Select the issue types you want the dashboard to treat as defects. This is used in the quality metric to indicate the effort spent fixing bugs and other defects.

## Make your data work for you

We hope the DevOps template has given you a valuable insight into your teams current engineering health, and sparked a few ideas for how you can use the data pipeline data to make better business decisions.

You should treat this template as a jumping off point. The way we have chosen to calculate the metrics may not suit the way your teams work. Take the responsiveness and productivity charts for example. Weve used an issue as the basic unit of measurement. This works great for Kanban teams, where we can expect issues to be of a similar size. However for a scrum team, you may find it better to use story points as the unit of measurement, so you can measure the average completion time per story point, rather than per issue.

Ready to get started?

- [Deploy the DevOps dashboard in Tableau](#)
- [Deploy the DevOps dashboard in PowerBI](#)

# Deploy the DevOps dashboard in Tableau

The [data pipeline](#) allows you to export data from your Jira instance for analysis in your favorite business intelligence tool.

To get you started, we've developed a DevOps template in Tableau which provides useful insights into the health of your engineering teams, and should provide a great jumping off point for creating your own dashboards and reports.

[Learn how to make the most of the data pipeline with the DevOps dashboard](#)

This page will guide you through how to deploy our sample DevOps template in Tableau Desktop, and connect it to your data source.

[Download the DevOps dashboard template for Tableau](#)

The template has been tested with PostgreSQL and Microsoft SQL Server. It requires Tableau Desktop 2021.1 or later.

## Import data pipeline CSVs into your database

Before you can use the template, you need to import the CSV files exported by the data pipeline in Jira Data Center into a database. You can also import the files directly into Tableau, but for this guide we'll assume you'll use an external database.

The sample DevOps template expects your database to contain the following tables:

- `issues` table containing data from the `issues_job<job_id>_<timestamp>.csv` file.
- `issue_history` table containing data from the `issue_history_job<job_id>_<timestamp>.csv` file.

## PostgreSQL example

In your database, create an `issues`, `issue_history` and `users` table as follows.

### On this page:

- [Import data pipeline CSVs into your database](#)
- [Launch the template and connect to your database](#)
- [Known issues and limitations](#)

```

CREATE TABLE issues (
  id varchar(50),
  instance_url varchar(1000),
  "key" varchar(1000),
  url varchar(1000),
  project_key varchar(1000),
  project_name varchar(1000),
  project_type varchar(1000),
  project_category varchar(1000),
  issue_type varchar(1000),
  summary varchar(1000),
  description varchar(2000),
  environment varchar(2000),
  creator_id varchar(50),
  creator_name varchar(1000),
  reporter_id varchar(50),
  reporter_name varchar(1000),
  assignee_id varchar(50),
  assignee_name varchar(1000),
  status varchar(1000),
  status_category varchar(1000),
  priority_sequence varchar(1000),
  priority_name varchar(1000),
  resolution varchar(1000),
  watcher_count varchar(50),
  vote_count varchar(50),
  created_date timestamp,
  resolution_date varchar(50),
  updated_date varchar(50),
  due_date varchar(50),
  estimate varchar(50),
  original_estimate varchar(50),
  time_spent varchar(50),
  parent_id varchar(50),
  security_level varchar(1000),
  labels varchar(1000),
  components varchar(1000),
  affected_versions varchar(1000),
  fix_versions varchar(100));

CREATE TABLE issue_history (
  issue_id varchar(50),
  changelog_id varchar(50),
  author_id varchar(50),
  author_key varchar(1000),
  created_date timestamp,
  field_type varchar(1000),
  field varchar(1000),
  "from" varchar(1000),
  from_string varchar(1000),
  "to" varchar(1000),
  to_string varchar(1000),
  additional_information varchar(2000));

CREATE TABLE users (
  user_id varchar(50),
  instance_url varchar(1000),
  user_name varchar(1000),
  user_fullname varchar(1000),
  user_email varchar(1000)
);

```

Import the appropriate CSV file into each table. You can adjust the above script and CSV methods as appropriate for other databases.

For PostgreSQL, there are several methods you can use. See [Import CSV File Into PostgreSQL Table](#) for some suggested methods.

Launch the template and connect to your database

Now that you have imported your data, you can launch the template in Tableau Desktop, and connect it to your data.

To launch the template and connect it to your database:

1. Download the [DevOps dashboard template for Tableau](#).
2. In Tableau Desktop 2021.1 or later, open the Atlassian DevOps Tableau packaged workbook. The template will be populated with some sample data.
3. Select **New Data Source**, and follow the prompts to connect to your database.
4. Drag the **New Custom SQL** placeholder to the canvas. If your database doesn't have Custom SQL option, create a view in your database using the custom SQL provided below, then drag that view to the canvas instead.
5. Connect with the following SQL query. Note that you will need to escape any reserved words using the appropriate escape characters for your database.

```

SELECT h.issue_id as h_issue_id
      ,h.changelog_id as h_changelog_id
      ,h.author_id as h_author_id
      ,h.author_key as h_author_key
      ,h.created_date as h_created_date
      ,h.field as h_field
      ,h.field_type as h_field_type
      ,h.from as h_from
      ,h.from_string as h_from_string
      ,h.to as h_to
      ,h.to_string as h_to_string
      ,h.additional_information as h_additional_information
,i.id as id
      ,i.instance_url
      ,i.key as key
      ,i.url
      ,i.project_key
      ,i.project_name
      ,i.project_type
      ,i.project_category
      ,i.issue_type
      ,i.summary
      ,i.description
      ,i.environment
      ,i.creator_id
      ,u1.user_name as creator_name
      ,i.reporter_id
      ,u2.user_name as reporter_name
      ,i.assignee_id
      ,u3.user_name as assignee_name
      ,i.status
      ,i.status_category
      ,i.priority_sequence
      ,i.priority_name
      ,i.resolution
      ,i.watcher_count
      ,i.vote_count
      ,i.created_date as created_date
      ,i.resolution_date
      ,i.updated_date
      ,i.due_date
      ,i.estimate
      ,i.original_estimate
      ,i.time_spent
      ,i.parent_id
      ,i.security_level
      ,i.labels
      ,i.components
      ,i.affected_versions
      ,i.fix_versions
FROM issue_history h
join issues i on h.issue_id = i.id
join users u1 on u1.user_id = i.creator_id
join users u2 on u2.user_id = i.reporter_id
join users u3 on u3.user_id = i.assignee_id
where h.field_type = 'jira' and h.field= 'status'

```

6. Once the connection is made, select **Dashboard** to start the extract process.
7. Once the extract has been created, go to any sheet and select **Data > Replace Data Source**.
8. Select your new data source name.
9. The new data source will now be used by all sheets.

To learn more about how to configure the dashboard, and the metrics used, go to [Make the most of the data pipeline with the DevOps dashboard](#).

### Good to know

- Don't change the connection type to Live. The dashboard is designed to be database agnostic, and some functions aren't supported by all databases. Keeping the connection as Extract keeps the calculations intact.
- Dashboard formatting will default to Tableau default colors if the connection type is changed.
- There are some reserved words in the Table names. The reserved words are from, to, key, status. If you change the connection type you will need to escape these reserved words using the appropriate escape characters for your database. A good approach would be to run the custom SQL query directly in the database interface to ensure the syntax of the SQL is correct.
- Not all database connections have a Custom SQL option. If your Database doesn't support Custom SQL you will need to create a view with the SQL provided. Creating a view ensures the columns are renamed, and we filter records as required by the dashboard.
- The dashboard was built to use an ODBC connection to make it database agnostic. Learn more about [ODBC connections in Tableau](#). If you choose to use the ODBC connection option, you will need to install the appropriate ODBC driver for your database, and then set up a data source name(DSN).

## Known issues and limitations

### **ODBC driver does not support all the capabilities used by Tableau warning**

When connecting to the database, Tableau may warn you that not all capabilities will be available with that driver. See [Tableau and ODBC](#) in the Tableau documentation for more information on these warnings.

# Deploy the DevOps dashboard in PowerBI

The [data pipeline](#) allows you to export data from your Jira instance for analysis in your favorite business intelligence tool.

To get you started, we've developed a DevOps template in Tableau which provides useful insights into the health of your engineering teams, and should provide a great jumping off point for creating your own dashboards and reports.

[Learn how to make the most of the data pipeline with the DevOps dashboard](#)

This page will guide you through how to deploy our sample DevOps template in Microsoft PowerBI, and connect it to your data source.

[Download the DevOps dashboard template for PowerBI](#)

## Import data pipeline CSVs

Before you can use the template with your own data, you need to import the CSV files exported by the data pipeline in Jira Data Center into a database or blob storage.

The sample DevOps template uses the following files:

- `issues_job<job_id>_<timestamp>.csv` file.
- `issue_history_job<job_id>_<timestamp>.csv` file.

How you import this data depends on where it will be stored.

### Azure blob storage

Upload the two CSV files into Container.

You will need to rename the files as follows:

- `issues.csv`
- `issue_history.csv`

### Azure SQL and PostgreSQL

In your database, create an `issues` and `issue_history` table as follows.

#### On this page:

- [Import data pipeline CSVs](#)
- [Launch the template and connect to your data](#)

```

CREATE TABLE issues (
  id varchar(50),
  instance_url varchar(1000),
  "key" varchar(1000),
  url varchar(1000),
  project_key varchar(1000),
  project_name varchar(1000),
  project_type varchar(1000),
  project_category varchar(1000),
  issue_type varchar(1000),
  summary varchar(1000),
  description varchar(2000),
  environment varchar(2000),
  creator_id varchar(50),
  creator_name varchar(1000),
  reporter_id varchar(50),
  reporter_name varchar(1000),
  assignee_id varchar(50),
  assignee_name varchar(1000),
  status varchar(1000),
  status_category varchar(1000),
  priority_sequence varchar(1000),
  priority_name varchar(1000),
  resolution varchar(1000),
  watcher_count varchar(50),
  vote_count varchar(50),
  created_date varchar(50),
  resolution_date varchar(50),
  updated_date varchar(50),
  due_date varchar(50),
  estimate varchar(50),
  original_estimate varchar(50),
  time_spent varchar(50),
  parent_id varchar(50),
  security_level varchar(1000),
  labels varchar(1000),
  components varchar(1000),
  affected_versions varchar(1000),
  fix_versions varchar(100));

CREATE TABLE issue_history (
  issue_id varchar(50),
  changelog_id varchar(50),
  author_id varchar(50),
  author_key varchar(1000),
  created_date varchar(50),
  field_type varchar(1000),
  field varchar(1000),
  "from" varchar(1000),
  from_string varchar(1000),
  "to" varchar(1000),
  to_string varchar(1000),
  additional_information varchar(2000));

```

Import the appropriate CSV file into each table.

For SQL, see [Load data from CSV into Azure SQL Database or SQL Managed Instance \(flat files\)](#) in the Microsoft documentation.

For PostgreSQL, there are several methods you can use. See [Import CSV File Into PostgreSQL Table](#) for some suggested methods.

## Launch the template and connect to your data

Now that you have imported your data, we can launch the .pbix template in PowerBI, and connect it to your data source.

To connect to your data:

1. Open PowerBI Desktop and launch the file.

2. Select **Get data**.
3. Choose your data source and follow the prompts to enter your database details and credentials.
4. Choose issues and issue\_history tables to include and select **Load**.
5. Select **Refresh** to update the dashboard data.
6. The dashboard should display your data.

For more information on connecting or replacing a data source, refer to the [Microsoft PowerBI documentation](#).

# Important directories and files

On this page:

- [Jira installation directory](#)
  - [Important files and directories](#)
    - `<jira-application-dir>/atlassian-jira/WEB-INF/classes/jira-application.properties`
    - `<jira-application-dir>/atlassian-jira/WEB-INF/classes/jpm.xml`
    - `<jira-application-dir>/atlassian-jira/WEB-INF/lib/`
    - `<jira-application-dir>/atlassian-jira/WEB-INF/classes/log4j.properties`
    - `<jira-application-dir>/atlassian-jira/WEB-INF/classes/entityengine.xml`
    - `<jira-application-dir>/`
    - `conf/server.xml`
    - `logs/atlassian-jira-gc-timestamp.log`
    - [Memory settings](#)
- [Jira home directory](#)
- [Important files](#)
  - `dbconfig.xml`
  - `jira-config.properties`
- [Important subdirectories](#)
  - `data`
  - `export`
  - `log`
  - `plugins`
  - `caches`
  - `tmp`

## Jira installation directory

The Jira installation directory is the directory into which the Jira application files and libraries have been extracted, either:

- by the [Windows](#) installer, or
- by the [Linux](#) installers

Jira does not modify or store any data in this directory.

## Important files and directories

 The directories/files described below are found under different sub-directories of the 'Jira Installation Directory', depending on whether you have installed a recommended Windows, Linux or Archive Jira. Please substitute the following directories for the `<Jira-application-dir>` placeholder (used throughout the rest of this section), as follows:

- **'Recommended' distributions** the `atlassian-jira` subdirectory of the 'Jira Installation Directory' installed using the '[Windows Installer](#)' and '[Linux Installer](#)', and there associated installations from archive files (.zip and tar.gz respectively).
- The default installation directory on Linux is:

```
/opt/atlassian/jira/
```

```
<jira-application-dir>/atlassian-jira/WEB-INF/classes/jira-application.properties
```

This file tells Jira where to find the [Jira application home directory](#).

 Be aware that your Jira home directory defined in this file can be overridden. See [Setting your Jira application home directory](#) for more information.

```
<jira-application-dir>/atlassian-jira/WEB-INF/classes/jpm.xml
```

This file stores the default values for [Jira's advanced configuration settings](#) and should not be modified. The default values of properties in this file are customized (i.e. overridden) by redefining them in either the `jira-config.properties` file (in your [Jira application home directory](#)) or the Jira database (via the Jira administration area). See [Advanced Jira configuration](#) for more information.

```
<jira-application-dir>/atlassian-jira/WEB-INF/lib/
```

This is the directory where plugins built on Atlassian's Plugin Framework 1 (i.e. 'Plugins 1' plugins) are stored. If you are [installing a new 'Plugins 1' plugin](#), you will need to deploy it into this directory. 'Plugins 2' plugins should be stored in the [Jira application home directory](#).

```
<jira-application-dir>/atlassian-jira/WEB-INF/classes/log4j.properties
```

Jira's logging configuration file. See [Logging and profiling](#).

The actual log files generated by Jira can be found in the following locations:

- **Jira application log** `bin/atlassian-jira.log`
- **Application server log** generally the application server log file can be found under the `logs` directory. However, this can vary depending on the application server you are running.

```
<jira-application-dir>/atlassian-jira/WEB-INF/classes/entityengine.xml
```

This file configures the OFBiz Entity Engine, which Jira uses to store persist data in a data source.

```
<jira-application-dir>/
```

This file includes garbage collection (GC) logs that are used to improve the performance of Jira applications. The log statements indicate when Java is collecting garbage, how long this process takes, and how much memory has been freed. The file is generated automatically, and is also included in the Support Zip package. For more info, see [Using garbage collection logs](#).

 The sub-directories/files described below are found under the root of the Jira application installation directory.

```
conf/server.xml
```

This file is used for Jira SSL configuration. See [Running Jira applications over SSL or HTTPS](#).

```
logs/atlassian-jira-gc-timestamp.log
```

These files include garbage collection (GC) logs that can be used to monitor the performance of Jira applications. The log statements indicate when Java is collecting garbage, how long this process takes, and which resources can be freed. The files are created automatically, and then overwritten if the maximum number of files (5) is reached. The timestamp indicates when the Jira session related to the logs was started. For more info, see [Using garbage collection logs](#).

### Memory settings

The file used to edit JAVA\_OPTS memory settings will depend on the method used to install Jira, as well as the operating system used for your installation.

For example, if you are running Jira on Tomcat in Windows (manual startup), you would update the following file:

```
bin\setenv.bat
```

whereas for Jira on Tomcat in Linux/Unix, you would update this file:

```
bin/setenv.sh
```

See [Increasing Jira memory](#) for further details.

## Jira home directory

The Jira home directory contains key data that help define how Jira works. This document outlines the purpose of the various files and subdirectories within the Jira home directory.

If Jira was installed using the automated [Windows](#) or [Linux](#) installers, the default location of the Jira home directory is:

- C:\Program Files\Atlassian\Application Data\JIRA (on Windows) or
- /var/atlassian/application-data/JIRA (on Linux)

If you install Jira from an archive file, the Jira home directory can be any suitable location that is accessible by your JIRA installation. Typical example locations might be:

- C:\jira\home (on Windows) or
- /var/jira-home (on Linux or Solaris)

⚠ However, avoid locating the Jira home directory inside the [Jira application installation directory](#).

✔ For information on specifying the location of the Jira home directory, please see [Setting your Jira application home directory](#).

## Important files

### `dbconfig.xml`

This file (located at the root of your Jira home directory) defines all details for Jira's database connection. This file is typically created by running the [Jira setup wizard](#) on new installations of Jira or by configuring a database connection using the [Jira configuration tool](#).

You can also create your own `dbconfig.xml` file. This is useful if you need to specify additional parameters for your specific database configuration, which are not generated by the setup wizard or Jira configuration tool. For more information, refer to the 'manual' connection instructions of the appropriate database configuration guide in [Connecting Jira to a database](#).

### `jira-config.properties`

This file (also located at the root of your Jira home directory) stores custom values for most of [Jira's advanced configuration settings](#). Properties defined in this file override the default values defined in the `jpm.xml` file (located in your [Jira application installation directory](#)). See [Advanced Jira configuration](#) for more information.

ℹ In new Jira installations, this file may not initially exist and if so, will need to be created manually. See [Making changes to the `jira-config.properties` file](#) for more information. This file is typically present in Jira installations upgraded from version 4.3 or earlier, whose [advanced configuration options](#) had been customized (from their default values).

## Important subdirectories

### `data`

This directory contains application data for your Jira instance, including attachments (for every version of each attachment stored in Jira).

### `export`

Jira will place its [automated backup archives](#) into this directory.

### `log`

Jira will place its logs into this directory. (Note: if the Jira home directory is not configured, then the logs will be placed into the current working directory instead).

The logs will only start showing up once the first log message is written to them. For example, the internal access log will not be created until Jira starts writing to it.

You can change the location of the log file using `log4j.properties` as described in the documentation on [Logging and profiling](#).

### **plugins**

This is the directory where plugins built on [Atlassian's Plugin Framework 2](#) (i.e. 'Plugins 2' plugins) are stored. If you are [installing a new 'Plugins 2' plugin](#), you will need to deploy it into this directory under the `installed-plugins` sub-directory.

'Plugins 1' plugins should be stored in the [Jira application installation directory](#).

This directory is created on Jira startup, if it does not exist already.

### **caches**

This is where Jira stores caches including:

- Lucene indexes - see [Troubleshoot index problems in Jira server](#)
- OSGi framework caches

These files are vital for Jira performance and should not be modified or removed externally while Jira is running.

See [Search indexing](#) for further details.

### **tmp**

Any temporary content created for various runtime functions such as exporting, importing, file upload and indexing is stored under this directory.

You can remove files from this directory while Jira is running, but we recommend that you shut down Jira first before altering the contents of this directory.

# Jira application installation directory

The Jira installation directory is the directory into which the Jira application files and libraries have been extracted, either:

- by the [Windows](#) installer, or
- by the [Linux](#) installers

Jira does not modify or store any data in this directory.

## Important files and directories

**i** The directories/files described below are found under different sub-directories of the 'Jira Installation Directory', depending on whether you have installed a recommended Windows, Linux or Archive Jira. Please substitute the following directories for the `<Jira-application-dir>` placeholder (used throughout the rest of this section), as follows:

- **'Recommended' distributions** the `atlassian-jira` subdirectory of the 'Jira Installation Directory' installed using the '[Windows Installer](#)' and '[Linux Installer](#)', and there associated installations from archive files (.zip and tar.gz respectively).
- The default installation directory on Linux is:

```
/opt/atlassian/jira/
```

```
<jira-application-dir>/atlassian-jira/WEB-INF/classes/jira-application.properties
```

This file tells Jira where to find the [Jira application home directory](#).

**!** Be aware that your Jira home directory defined in this file can be overridden. See [Setting your Jira application home directory](#) for more information.

```
<jira-application-dir>/atlassian-jira/WEB-INF/classes/jpm.xml
```

This file stores the default values for [Jira's advanced configuration settings](#) and should not be modified. The default values of properties in this file are customized (i.e. overridden) by redefining them in either the `jira-config.properties` file (in your [Jira application home directory](#)) or the Jira database (via the Jira administration area). See [Advanced Jira configuration](#) for more information.

```
<jira-application-dir>/atlassian-jira/WEB-INF/lib/
```

This is the directory where plugins built on Atlassian's Plugin Framework 1 (i.e. 'Plugins 1' plugins) are stored. If you are [installing a new 'Plugins 1' plugin](#), you will need to deploy it into this directory. 'Plugins 2' plugins should be stored in the [Jira application home directory](#).

```
<jira-application-dir>/atlassian-jira/WEB-INF/classes/log4j.properties
```

Jira's logging configuration file. See [Logging and profiling](#).

The actual log files generated by Jira can be found in the following locations:

- **Jira application log** `bin/atlassian-jira.log`
- **Application server log** generally the application server log file can be found under the `logs` directory. However, this can vary depending on the application server you are running.

```
<jira-application-dir>/atlassian-jira/WEB-INF/classes/entityengine.xml
```

This file configures the OFBiz Entity Engine, which Jira uses to store persist data in a data source.

```
<jira-application-dir>/
```

This file includes garbage collection (GC) logs that are used to improve the performance of Jira applications. The log statements indicate when Java is collecting garbage, how long this process takes, and how much memory has been freed. The file is generated automatically, and is also included in the Support Zip package. For more info, see [Using garbage collection logs](#).

 The sub-directories/files described below are found under the root of the Jira application installation directory.

`conf/server.xml`

This file is used for Jira SSL configuration. See [Running Jira applications over SSL or HTTPS](#).

`logs/atlassian-jira-gc-timestamp.log`

These files include garbage collection (GC) logs that can be used to monitor the performance of Jira applications. The log statements indicate when Java is collecting garbage, how long this process takes, and which resources can be freed. The files are created automatically, and then overwritten if the maximum number of files (5) is reached. The timestamp indicates when the Jira session related to the logs was started. For more info, see [Using garbage collection logs](#).

### Memory settings

The file used to edit JAVA\_OPTS memory settings will depend on the method used to install Jira, as well as the operating system used for your installation.

For example, if you are running Jira on Tomcat in Windows (manual startup), you would update the following file:

`bin\setenv.bat`

whereas for Jira on Tomcat in Linux/Unix, you would update this file:

`bin/setenv.sh`

See [Increasing Jira memory](#) for further details.

# Jira application home directory

The Jira home directory contains key data that help define how Jira works. This document outlines the purpose of the various files and subdirectories within the Jira home directory.

If Jira was installed using the automated [Windows](#) or [Linux](#) installers, the default location of the Jira home directory is:

- C:\Program Files\Atlassian\Application Data\JIRA (on Windows) or
- /var/atlassian/application-data/JIRA (on Linux)

If you install Jira from an archive file, the Jira home directory can be any suitable location that is accessible by your JIRA installation. Typical example locations might be:

- C:\jira\home (on Windows) or
- /var/jira-home (on Linux or Solaris)

⚠ However, avoid locating the Jira home directory inside the [Jira application installation directory](#).

✓ For information on specifying the location of the Jira home directory, please see [Setting your Jira application home directory](#).

## Important files

### `dbconfig.xml`

This file (located at the root of your Jira home directory) defines all details for Jira's database connection. This file is typically created by running the [Jira setup wizard](#) on new installations of Jira or by configuring a database connection using the [Jira configuration tool](#).

You can also create your own `dbconfig.xml` file. This is useful if you need to specify additional parameters for your specific database configuration, which are not generated by the setup wizard or Jira configuration tool. For more information, refer to the 'manual' connection instructions of the appropriate database configuration guide in [Connecting Jira to a database](#).

### `jira-config.properties`

This file (also located at the root of your Jira home directory) stores custom values for most of [Jira's advanced configuration settings](#). Properties defined in this file override the default values defined in the `jpm.xml` file (located in your [Jira application installation directory](#)). See [Advanced Jira configuration](#) for more information.

ℹ In new Jira installations, this file may not initially exist and if so, will need to be created manually. See [Making changes to the `jira-config.properties` file](#) for more information. This file is typically present in Jira installations upgraded from version 4.3 or earlier, whose [advanced configuration options](#) had been customized (from their default values).

## Important subdirectories

### `data`

This directory contains application data for your Jira instance, including attachments (for every version of each attachment stored in Jira).

### `export`

Jira will place its [automated backup archives](#) into this directory.

### `log`

Jira will place its logs into this directory. (Note: if the Jira home directory is not configured, then the logs will be placed into the current working directory instead).

The logs will only start showing up once the first log message is written to them. For example, the internal access log will not be created until Jira starts writing to it.

You can change the location of the log file using `log4j.properties` as described in the documentation on [Logging and profiling](#).

#### **plugins**

This is the directory where plugins built on [Atlassian's Plugin Framework 2](#) (i.e. 'Plugins 2' plugins) are stored. If you are [installing a new 'Plugins 2' plugin](#), you will need to deploy it into this directory under the `installed-plugins` sub-directory.

'Plugins 1' plugins should be stored in the [Jira application installation directory](#).

This directory is created on Jira startup, if it does not exist already.

#### **caches**

This is where Jira stores caches including:

- Lucene indexes - see [Troubleshoot index problems in Jira server](#)
- OSGi framework caches

These files are vital for Jira performance and should not be modified or removed externally while Jira is running.

See [Search indexing](#) for further details.

#### **tmp**

Any temporary content created for various runtime functions such as exporting, importing, file upload and indexing is stored under this directory.

You can remove files from this directory while Jira is running, but we recommend that you shut down Jira first before altering the contents of this directory.

# Setting your Jira application home directory

The [Jira home directory](#) contains key data that helps define how Jira works. You must have a home directory specified for your Jira instance before you can start it.

What location should I specify?

- You can choose any location on the disk, but make sure to use an **absolute path**.
- You can't use the same home directory for multiple Jira instances. Each instance must have its own home directory.
- Locate the home directory independently of the installation directory (don't nest one within the other). This will minimize information being lost during major operations, like backing up or restoring.

## Setting Jira home directory

### Environment variable RECOMMENDED

We recommend setting Jira home through the `JIRA_HOME` environment variable, because it takes precedence over any other method. For example, if you have different paths specified in the environment variable and the properties file, Jira will use the variable.

Use one of the following approaches:

- Enter the following command at a shell/console prompt before running Jira.

```
export JIRA_HOME=/path/to/jira/home
```

- Add the above command to the `/bin/start-jira.sh` script.

Use one of the following approaches:

- Configure the `JIRA_HOME` environment variable through the Windows user interface, typically through 'My Computer' or 'Computer'.
- At the command prompt, enter the following command:

```
set JIRA_HOME=C:\path\to\jira\home
```

- Add the above command to the `\bin\start-jira.bat` file.

### Properties file

Add the path to the Jira home directory to the `<installation-directory>/atlassian-jira/WEB-INF/classes/jira-application.properties` file, like in the following example:

```
jira.home = /Users/charlie/Jira/jira_home
```

#### Notes for Windows

- Use double back-slashes (`\`) between subdirectories, for example `C:\path\to\Jira\home`.
- If you define a UNC path, make sure to double escape the leading backslash, for example `\\machinename\path\to\JIRA\home`.

### Configuration tool

You can also change the location of your home directory by using the [Jira configuration tool](#).

### Alternative method

Alternatively, you can specify the location of your Jira home directory as property within your application server. This method would override the `jira.home` specified in the properties file.

- Configure a new web context property called '`jira.home`' for your application server.

To do this, you need to define this web context property inside a `<parameter/>` element (as a child of the `<context/>` element) in your `server.xml` file.

```
<Context ...>
...
  <Parameter name="jira.home" value="c:/jira/home"/>
...
</Context>
```

### Notes

- If you're using the Windows installer, you don't need to configure the Jira home directory separately, as you will be prompted to specify this location during the installation process.
- The Jira installer may not be able to create the home due to permission problems. If this is the case, see [Jira is unable to start due to 'Could not create necessary subdirectory'](#)

# Integrating Jira applications with a Web server

The following pages contain information on integrating Jira applications with a web server.

- [Integrating Jira applications with IIS](#)
- [Integrating Jira with Apache](#)

# Integrating Jira applications with IIS

 The content on this page relates to platforms that are not supported by Jira. Consequently, Atlassian **can not guarantee providing any support for it**. Please be aware that this material is provided for your information only, and using it is done so at your own risk.

## Ghostcat vulnerability in Apache Tomcat

Following this guide might make your Jira instance prone to a recent high-risk vulnerability found in Apache Tomcat.

We recommend that you wait until Jira is bundled with the Tomcat version that fixes this issue, we'll update this note once its released. For more info about this vulnerability, see:

- [CVE-2020-1938: Ghostcat - Apache Tomcat AJP File Read/Inclusion Vulnerability](#)

Until then, if you need to use the AJP Connector, there are steps you can take to mitigate this issue. For more info, see [this article](#).

This page describes how to configure Microsoft's IIS web server and Jira such that IIS forwards requests on to Jira, and responses back to the user. This is useful if you already have IIS running serving web pages (e.g. <http://mycompany.com>), and wish to integrate Jira as just another URL (e.g. <http://mycompany.com/jira>).

Jira is written in Java, and needs a Java Application Server (servlet container) to run. As IIS does not provide services of a Java Application Server, it is not possible to deploy Jira directly into IIS. It is possible, however, to configure IIS to proxy requests for Jira to an application server where Jira is deployed. Therefore, if your main website is running in IIS, it is possible to integrate Jira into this website.

If you need to integrate Jira with IIS, Jira needs to be deployed into a Java application server (such as [Apache Tomcat](#)), which provides IIS integration capability.

If you are running Jira against an application server other than Apache Tomcat, please consult that application server's documentation to determine whether it is possible (and how) to integrate the application server with IIS.

To integrate Jira with IIS you will need to:

1. [Configure Jira and test that it works on its own](#)
2. [Configure Tomcat to accept proxied requests from IIS](#)
3. [Configure IIS to forward Jira requests to Tomcat](#)

## 1. Configure Jira

1. Follow [the Jira installation guide](#) to install and configure Jira. Note that Jira can be installed on the same machine as IIS, but this is not necessary.
2. Change the context path of the Jira web application:  
To allow IIS to proxy requests to Jira, Jira web application must be deployed with a context path (e.g. the **jira** in <http://localhost:8080/jira> ([http://localhost:8080\\*/jira\\*](http://localhost:8080*/jira*))) in Tomcat. The context path **must** be set to the path in the URL that IIS will use to proxy requests. For example, if your website is running with address [www.example.com](http://www.example.com) in IIS, and you would like to make Jira available under [www.example.com/jira](http://www.example.com/jira), you will need to set Jira's context path to `"/jira"` in Tomcat.  
To do this, edit the `conf/server.xml` file. Change the `path` attribute of the `Context` element to `"/jira"`.
3. Restart Jira after changing the context path.
4. Set the **'Base URL'** to include the context path (see [Configuring Jira options](#)).
5. Turn Jira's GZip compression **OFF** (since there will be no benefit from GZip compression once proxying is implemented).
6. Test that Jira works correctly by pointing your web browser directly at Tomcat (e.g. <http://localhost:8080/jira>) and going through Jira's Setup Wizard. If you have completed the Setup Wizard previously, try creating an issue or editing one. Please ensure that no errors occur.

## 2. Configure Tomcat to accept proxied requests

### HTTP/1.1 Connector

If you are using the HTTP/1.1 Connector, you will need to add the following attributes to the Connector port in Tomcat's `server.xml`:

```
proxyName="mycompany.com" proxyPort="80"
```

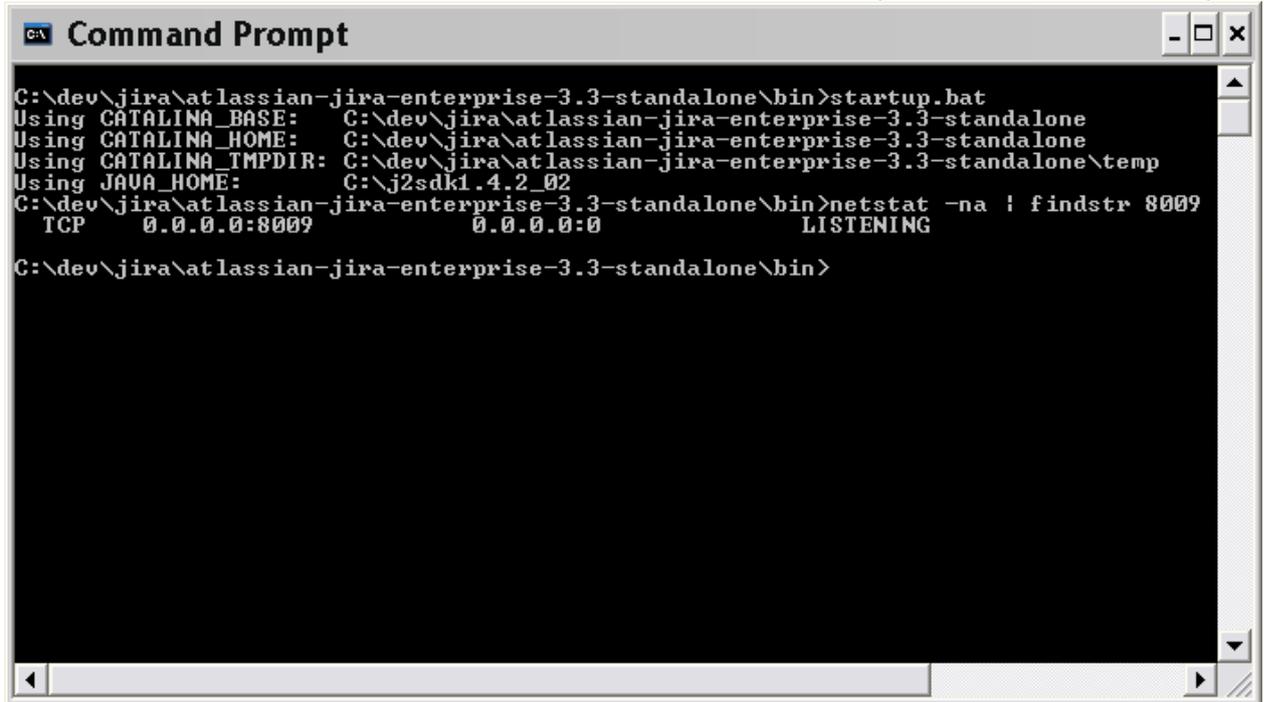
Please refer to the [Integrating Jira with Apache](#) for reference.

1. Enable **AJP/1.3 Connector** in Tomcat: To allow Tomcat to accept requests for Jira from IIS, edit the `conf/server.xml` file and ensure that the **AJP/1.3 Connector** is enabled (i.e. *not* commented out). To enable the AJP/1.3 Connector in a Jira remove the comment symbols around the following section in the `conf/server.xml` file:

```
<Connector port="8009" enableLookups="false" redirectPort="8443" protocol="AJP/1.3" />
```

The above example configures Tomcat to listen for proxied IIS requests on port 8009. If this port is already in use on the machine where Jira is running, please change to another port.

2. Restart Tomcat and ensure that no errors regarding used ports appear in the logs or in the Tomcat Console.
3. Ensure that the AJP Connector is listening on the specified port (8009 by default). One way to do this is to use the `"netstat -na"` command in the command window and see if port 8009 is listed in the output:



```

C:\dev\jira\atlassian-jira-enterprise-3.3-standalone\bin>startup.bat
Using CATALINA_BASE:   C:\dev\jira\atlassian-jira-enterprise-3.3-standalone
Using CATALINA_HOME:   C:\dev\jira\atlassian-jira-enterprise-3.3-standalone
Using CATALINA_TMPDIR: C:\dev\jira\atlassian-jira-enterprise-3.3-standalone\temp
Using JAVA_HOME:       C:\j2sdk1.4.2_02
C:\dev\jira\atlassian-jira-enterprise-3.3-standalone\bin>netstat -na | findstr 8009
TCP        0.0.0.0:8009          0.0.0.0:0           LISTENING
C:\dev\jira\atlassian-jira-enterprise-3.3-standalone\bin>

```

## 3. Configure IIS to forward requests to Jira

On the machine where IIS is deployed:

1. Download the ISAPI Redirect DLL from the [Apache site](#). When downloading, choose the version of Windows that IIS is running on (either win32 or win64), and then **choose the latest available jk version**.

 The file to download is named `isapi_redirect_X.X.X.dll`, where 'X.X.X' is the version number. You will need to remove the version number from the DLL file (i.e. it needs to be named `isapi_redirect.dll`).

2. Place the DLL and the associated properties files in an installation directory. For the purpose of this document, we will assume the directory is `C:\tomcat_iis_connector`. Place the **isapi\_redirect.dll** in this directory. Then download the [isapi\\_redirect.properties](#) file and place this in the same directory as the **isapi\_redirect.dll** file.
3. Create a directory called 'conf' in your installation directory (`C:\tomcat_iis_connector\conf`). Download the files [uriworkermap.properties](#) and [workers.properties.minimal](#) and place them in the `C:\tomcat_iis_connector\conf` directory.
4. Create a directory called 'logs' (`C:\tomcat_iis_connector\logs`). This is where the logs associated with the **isapi\_redirect.dll** execution will be placed.
5. In the "`C:\tomcat_iis_connector`" directory you may need to modify the `isapi_redirect.properties` file. The `isapi_redirect.properties` file tells the connector where to find its configuration files and where the DLL can be found in relation to the IIS server. There are 5 properties in this file:
  - a. `extension_uri` the path to the virtual directory that contains the **isapi\_redirect.dll**
  - b. `log_file` the path to write the log file to
  - c. `log_level` the level at which the logs should be generated
  - d. `worker_file` the path to your `workers.properties.minimal` file in your installation
  - e. `worker_mount_file` the path to your `uriworkermap.properties` file in your installation.
 If you are installing the connector in `C:\tomcat_iis_connector` and you follow the instructions below about setting up the virtual directory for the **isapi\_redirect.dll**, then you should not have to change any properties in the provided file.
6. In the "`C:\tomcat_iis_connector\conf`" directory you may need to modify the `uriworkermap.properties` and the `workers.properties.minimal` files.

 The provided files contain the changes mentioned here and should work if you completely follow this document. **If you have deviated from this document, then you will need to modify these files as described below.**

The `workers.properties.minimal` file tells IIS where (IP address and port) Tomcat is running. The `uriworkermap.properties` tells IIS what requests to proxy to Tomcat.

To edit these files:

- a. Edit the `uriworkermap.properties` and ensure that it contains the following mapping for Jira. You do not need any other mappings.

```
/jira/*=worker1
```

 The mapping (e.g. `/jira/`) **must** be the same as the context path that Jira has been deployed with in Tomcat as described in the [Configure Jira](#) section of this document.

- b. Edit the `workers.properties.minimal` file and modify the `worker.ajp13w.host` property if necessary. This property should be set to the host name or the IP address of the machine where Tomcat (with Jira) is running. If Tomcat is running on the same machine as IIS then you can leave the property set to `localhost`. If you have specified a host name as the value of this property, please ensure that the IIS machine can correctly resolve it to the appropriate IP address.
- c. If you have modified the port for the AJP Connector you will need to modify the `worker.ajp13w.port` property. Here is an example of the file with Tomcat running on the same machine as IIS and using the default port (8009) for AJP:

```
worker.list=worker1

#
# Defining a worker named worker1 and of type ajp13.
# Note that the name and the type do not have to match.
#
worker.worker1.type=ajp13
worker.worker1.host=localhost
worker.worker1.port=8009
```

7. Open **Control Panel**, then **Administrative Tools** and open **Internet Information Services**.

8. **IIS 7.0 only:** If you are using **IIS 7.0**, you will need to install two required service roles, ISAPI Extensions and ISAPI Filters:
  - a. Navigate to Start Menu > All Programs > Administration Tools > Service Manager.
  - b. Select 'Web Server (IIS)' in Server Manager > Roles.
  - c. Click 'Add Role Services' and follow the Wizard.
9. Add an **ISAPI Filter** to IIS, as described below:
  - **IIS 6.0 or earlier:**
    - a. Right-click on **Default Web Site** (or the Web Site that should be responsible for proxying requests to Jira), and click on **Properties**.
    - b. Click the **ISAPI Filters** tab.
    - c. Check if there is a Filter that points to the `isapi_redirect.dll` file and that it is in the right location. If not, click **Add** and create one. Enter `tomcat` as the Filter Name and enter the location of the `isapi_redirect.dll` file for the executable.
    - d. Click **Apply** and then **OK**.
  - **IIS 7.0:**
    - a. Click the **Default Web Site** (or the Web Site that should be responsible for proxying requests to Jira), and click on **ISAPI Filters**.
    - b. Click the **ISAPI Filters** icon.
    - c. Check if there is a Filter that points to the `isapi_redirect.dll` file and that it is in the right location. If not, click **Add** and create one. Enter `tomcat` as the Filter Name and enter the location of the `isapi_redirect.dll` file.
    - d. Click **OK**.
10. Create a **virtual directory** for Jira in IIS.
  - a. Right-click on **Default Web Site** (or the Web Site that should be responsible for proxying requests to Jira), choose **New** and then **Virtual Directory**.
  - b. Go through the creation wizard. Set the `alias` as the value of the Context Path (without slashes) that was set in the [Configure Jira](#) section of this document (see above). In our example this is `jira`.
  - c. This can point to any directory.
  - d. Complete the wizard.

 The reason for creating a virtual directory is so that requests without the trailing slash still work. For example, if you are deploying Jira under <http://www.example.com/jira/> without the virtual directory, then requests to <http://www.example.com/jira> will fail.

11. Create a **virtual directory** for access to the `isapi_redirect.dll` in IIS, as described below:
  - **IIS 6.0 or earlier:**
    - a. Right-click on **Default Web Site** (or the Web Site that should be responsible for proxying requests to Jira), choose **New** and then **Virtual Directory**.
    - b. Go through the creation wizard. Set the `alias` to be `jakarta`.
    - c. This must point to the directory in which the `isapi_redirect.dll` is installed. In our example this is `C:\tomcat_iis_connector`.
    - d. Complete the wizard, making sure that you grant the 'Execute' permission for the **Virtual Directory** by checking the 'Execute' checkbox.
  - **IIS 7.0:**
    - a. Right-click on **Default Web Site** (or the Web Site that should be responsible for proxying requests to Jira), and choose **Add Virtual Directory**.
    - b. Set the `alias` to be `jakarta`.
    - c. **Physical Path** must point to the directory in which the `isapi_redirect.dll` is installed. In our example this is `C:\tomcat_iis_connector`.
    - d. Click the 'jakarta' Virtual Directory and double-click 'Handler Mappings'.
    - e. Click 'Edit Feature Permissions' in the Action panel on the right-hand side.
    - f. Check the 'Execute' permission checkbox.

 This Virtual Directory is needed for the connector to work. The alias that you give the directory needs to be the same as the path set in the `isapi_redirect.properties` file, `extension_uri` property. In our example this value is: `/jakarta/isapi_redirect.dll`.

12. If using IIS 6.0 or 7.0, you will need to add the dll as a **Web Service Extension**, as described below.

- **IIS 6.0:**
    - a. Right-click on **Web Service Extensions** and choose **Add a new Web Service Extension...**
    - b. Enter `tomcat` for the **Extension Name** and then add the `isapi_redirect.dll` file to the required files.
    - c. Select the **Set extension status to Allowed** checkbox, then click **OK**.
  - **IIS 7.0:**
    - a. Navigate to the servers and highlight your server.
    - b. Navigate to 'ISAPI and CGI Restrictions'.
    - c. Add and allow the `isapi_redirect.dll` extension.
13. You will need to restart the IIS Service. To do this, browse to **Control Panel**, click **Administrative Tools**, click on **Services**, find the IIS Admin Service and click **restart**.
  14. You are done! To test the configuration, point your web browser at IIS and append Jira's context path to the URL. For example, if your website is running under the address of <http://www.example.com> and you have deployed Jira with the context path of `jira`, point your browser at <http://www.example.com/jira>.

## Troubleshooting

- **Whenever I go to Jira in my browser, a login panel pops up. I enter a valid username and password for Jira, but the panel pops up again.** Make sure that you have Anonymous Access set on the `jiravirtual` directory in IIS. It will be set to that if you have followed the above instructions. To check this:
  1. In 'Internet Information Services', right click the `jira` virtual directory and choose 'Properties'.
  2. Click the 'Directory Security' tab.
  3. Click the 'Edit...' button in the 'Anonymous access and authentication control' section.
  4. Make sure that the 'Anonymous access' tick box is selected, and make sure that nothing is selected in the 'Authenticated access' section. Do not select 'Basic authentication'. Do not select 'Integrated Windows authentication'.
- **Whenever I go to Jira in Internet Explorer, a login panel pops up. I enter a valid username and password for Jira, but the panel pops up again. This doesn't happen, however, in another browser such as Firefox or Safari. I can successfully log in to Jira in those browsers.** Make sure that you have Internet Explorer's User Authentication set to **Anonymous login**. To check this:
  1. In Internet Explorer, click the 'Tools' menu and select 'Internet Options'.
  2. Click the 'Security' tab.
  3. Select the security zone that the Jira server is in.
  4. Click the 'Custom level...' button.
  5. Scroll right down to the bottom to the 'User Authentication' section.
  6. Select 'Anonymous logon' (if it is not already selected).
  7. Click the 'OK' button on this screen, and again on the next screen.
  8. Restart Internet Explorer.
- **When I try to navigate to my Jira instance at <http://localhost/jira> in my browser, it prompts me to download a file with nonsensical information, rather than showing me my Jira instance.** Make sure that you have granted the 'Execute' permission to your Virtual Directory for Jira in IIS. See step 11 of the '3. Configure IIS to forward requests to Jira' section in this document for detailed instructions.

## Known issues

- **64 bit IIS:** If you are running a 64 bit OS, please use a [64 bit version of the Tomcat IIS connector](#).
- **Customer submitted solution:** If you must use a 32 bit IIS connector, you can do so by clicking `Application Pools > Advanced Settings > Allow 32bit applications`.
- **Customer submitted solution:** You need to set the ISAPI extension on the website.

# Integrating Jira with Apache

When opened in a viewport, the user will be redirected to: [Proxying Atlassian server applications with Apache HTTP Server \(mod\\_proxy\\_http\)](#).

 Atlassian applications allow the use of reverse-proxies within our products, however Atlassian Support does not provide assistance for configuring them. Consequently, Atlassian **can not guarantee providing any support for them**.

If assistance with configuration is required, please raise a question on [Atlassian Community](#).

This page describes how to integrate [Apache HTTP Server](#) (also referred to as `httpd`) with Jira, utilizing `mod_proxy` so that Apache operates as a reverse-proxy over HTTP. If HTTPS configuration is required, please see our [Integrating Jira with Apache using SSL](#) documentation. Configuring Apache allows for running Jira on non-standard HTTP port (such as 8080) and users will be able to access Jira over standard HTTP as their traffic will be routed through the proxy.

Apache can be configured to allow access to Jira in any of the following methods:

- Directly on its own domain: <http://jira.com>
- As a subdomain of another domain: <http://jira.atlassian.com>
- It can also be accessed on a context path on either a domain or subdomain: <http://atlassian.com/jira>

This documentation will cover a straightforward implementation of `mod_proxy` using the above three configurations. If a more complication solution is required, refer to the [Apache HTTP Server Version Documentation](#), consult with the Apache SME within your organization, and if need be raise a question on [Atlassian Answers](#), or get in touch with one of our [Atlassian Experts](#).

1. Jira is running on port 8080 on a server within the LAN that cannot be accessed externally (the router/firewall is not forwarding port 8080 to it).
2. Apache is set up on another server (or the same server as Jira) that can be accessed externally on HTTP (80).
3. Apache is then accessed over HTTP on the appropriate URL (`virtualHost`), routing the traffic to and from the Jira server.

## Step 1: Configure Tomcat

1. Stop Jira.
2. Edit Tomcat's `server.xml` to include the required Jira context path. The below example uses `path="jira"` - this means Jira is accessible on <http://jiraserver:8080/jira> given the default Jira port is used.

 This step is only required if Jira will be accessed on a context path, for example <http://atlassian.com/jira>. If this is not required, this step can be skipped.

### On this page:

- [Step 1: Configure Tomcat](#)
- [Step 2: Configure Apache HTTP Server](#)
  - [2.1 Enable the Proxy Modules](#)
  - [2.2. Configure Apache to use those Modules](#)
- [Step 3: Configure Jira](#)
- [Troubleshooting](#)
- [See also](#)

```

<Engine defaultHost="localhost" name="Catalina">
  <Host appBase="webapps" autoDeploy="true" name="localhost" unpackWARs="true">
    <Context docBase="{catalina.home}/atlassian-jira" path="/jira" reloadable="
false" useHttpOnly="true">

      <!--

=====
      Note, you no longer configure your database driver or connection parameters
here.

      These are configured through the UI during application setup.

=====
      -->
    <Resource auth="Container" factory="org.objectweb.jotm.
UserTransactionFactory" jotm.timeout="60" name="UserTransaction" type="javax.transaction.
UserTransaction"/>
    <Manager pathname="" />
  </Context>
</Host>

```

ⓘ Ensure the path value is set with a prepending forward slash (/). For example, path="/jira" rather than path="jira".

3. Edit Tomcat's server.xml to include a separate connector to proxy the requests. This requires the proxyName & proxyPort attributes. Replace them with the appropriate domain and port of the proxy, as in the below example:

```

<Service name="Catalina">

  <!-- Apache Proxy Connector -->
  <Connector acceptCount="100" connectionTimeout="20000" disableUploadTimeout="true"
enableLookups="false" maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25" port="8080"
protocol="HTTP/1.1" redirectPort="8443" useBodyEncodingForURI="true"
  proxyName="jira.atlassian.com" proxyPort="80" />

  <!-- Standard HTTP Connector -->
  <Connector acceptCount="100" connectionTimeout="20000" disableUploadTimeout="true"
enableLookups="false" maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25" port="8081"
protocol="HTTP/1.1" redirectPort="8443" useBodyEncodingForURI="true" />

```

4. Start Jira.
5. Test that Jira is accessible on the normal connector, using a context path if applicable - for example <http://jiraserver:8081/jira>.
6. Test that the new connector is in effect by accessing Jira on the appropriate proxy connector. The behavior varies depending on the context path:
  - a. If the context path is empty or root (/), visiting Jira via the proxy connector (e.g. <http://jiraserver:8080/>) should take you to Jira with a warning:

We've detected a potential problem with JIRA's Dashboard configuration that your administrator can correct. [Hide](#)

**Dashboard Diagnostics: Mismatched URL Hostname**

JIRA is reporting that it is running on the hostname 'jira.atlassian.com', which does not match the hostname used to run these diagnostics, 'localhost'. This is known to cause JIRA to construct URLs using the incorrect hostname, which will result in errors in the dashboard, among other issues.

The most common cause of this is the use of a reverse-proxy HTTP server (often Apache or IIS) in front of the application server running JIRA. While this configuration is supported, some additional setup might be necessary in order to ensure that JIRA detects the correct hostname.

The following articles describe the issue and the steps you should take to ensure that your web server and app server are configured correctly:

- [Gadgets do not display correctly after upgrade to JIRA 4.0](#)
- [Integrating JIRA with Apache](#)
- [Integrating JIRA with Apache using SSL](#)

If you believe this diagnosis is in error, or you have any other questions, please contact Atlassian Support.

**Detailed Error**  
[Click here to learn more](#)

- b. If the context path is other than empty or root(/), e.g. /jira, visiting Jira via the proxy connector (e.g. <http://jiraserver:8080/jira>) should redirect you to the configured proxy (e.g. <https://jira.atlassian.com/jira>).

## Step 2: Configure Apache HTTP Server

The installation of Apache and configuration of a DNS is not covered in this documentation. Additionally, it is assumed that Apache 2.2 has been installed and DNS entries have been configured for the Jira domain. As Apache's configuration is specific to the operation system that is used, only some distributions and their configurations are currently documented.

## 2.1 Enable the Proxy Modules

Debian/Ubuntu

1. Enable the module with the following:

```
$ sudo a2enmod proxy_http
Considering dependency proxy for proxy_http:
Enabling module proxy.
Enabling module proxy_http.
To activate the new configuration, you need to run:
  service apache2 restart
```

2. Restart Apache.

Windows/Other OS

1. Locate and edit the `httpd.conf` file, adding the below lines:

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_http_module modules/mod_proxy_http.so
```

2. Restart Apache.

## 2.2. Configure Apache to use those Modules

Debian/Ubuntu

1. Switch into user `root`.
2. Backup the existing instance or create a new one. Creating a new instance is not covered within this documentation (copying the default should be sufficient).
3. Modify the existing instance within `$APACHE_INSTALL/sites-available`, for example `default`.
4. Add the following inside the `VirtualHost`, replacing `jiraserver` with the hostname of the Jira server and also modifying the port if required.

On its own domain or subdomain:

```
# Jira Proxy Configuration:
<Proxy *>
    Order deny,allow
    Allow from all
</Proxy>

ProxyRequests            Off
ProxyPreserveHost        On
ProxyPass                 /          http://jiraserver:8080/
ProxyPassReverse          /          http://jiraserver:8080/
```

**i** Missing a forward slash at the end of the URL will cause proxy errors - ensure this is in place!

Using a context path:

```
# Jira Proxy Configuration:
<Proxy *>
    Order deny,allow
    Allow from all
</Proxy>

ProxyRequests          Off
ProxyPreserveHost      On
ProxyPass               /jira          http://jiraserver:8080/jira
ProxyPassReverse        /jira          http://jiraserver:8080/jira
```

**i** The path used must be identical to the Tomcat context path. For example, forwarding `/jira` to `/jira520` cannot be done without considerable rewrite rules that are not always reliable.

5. (Optional): Enable the instance with the following:

```
# a2ensite jira
Enabling site jira.
To activate the new configuration, you need to run:
    service apache2 reload
```

**i** This is only required if a new instance has been created in favor of using the default.

6. Reload the Apache configuration.
7. Test by accessing Jira through Apache, for example <http://jira.com> or <http://atlassian.com/jira>.

#### Windows/Other OS

1. Locate and edit the `httpd.conf` file.
2. Add the following inside the `VirtualHost`, replacing `jiraserver` with the hostname of the Jira server and also modifying the port if required.

#### On its own domain or subdomain:

```
# Jira Proxy Configuration:
<Proxy *>
    Order deny,allow
    Allow from all
</Proxy>

ProxyRequests          Off
ProxyPreserveHost      On
ProxyPass               /          http://jiraserver:8080/
ProxyPassReverse        /          http://jiraserver:8080/
```

**i** Missing a forward slash at the end of the URL will cause proxy errors - ensure this is in place!

#### Using a context path:

```
# Jira Proxy Configuration:
<Proxy *>
    Order deny,allow
    Allow from all
</Proxy>

ProxyRequests          Off
ProxyPreserveHost      On
ProxyPass               /jira          http://jiraserver:8080/jira
ProxyPassReverse        /jira          http://jiraserver:8080/jira
```

**i** The path used must be identical to the Tomcat context path. For example, forwarding `/jira` to `/jira520` cannot be done without considerable rewrite rules that are not always reliable.

3. Restart Apache.
4. Test by accessing Jira through Apache, for example <http://jira.com> or <http://atlassian.com/jira>.

### Step 3: Configure Jira

1. Set **Use gzip compression** to **OFF** as in [Configuring Jira options](#). GZIP compression is known to cause performance issues using a reverse-proxy, especially if the proxy is also compressing the traffic.
2. Set the **Base URL** to be the FQDN that Jira will be accessed on, for example <http://jira.atlassian.com>. This is also located in [Configuring Jira options](#).  
 ⚠️ Jira can only be configured to respond to a single URL and the Base URL (as in [Configuring Jira options](#)) must match the URL end-users are accessing. Misconfiguration of this may cause significant problems within Jira such as the Activity Stream and Dashboard Gadgets failing to function correctly.
3. Test by accessing Jira on the FQDN (e.g.: <http://jira.atlassian.com>), ensuring that Jira is accessible and all dashboard gadgets correctly display.

### Troubleshooting

- **Hijacked Sessions:** Some users have reported problems with user sessions being hijacked when the `mod_cache` module is enabled. If these problems are encountered, try disabling the `mod_cache` module.  
 ⓘ This module is enabled by default in some Apache HTTP Server version 2 distributions.
- **Permission Denied Errors enabling `mod_proxy` (and `mod_jk`) on Linux distros that use SELinux:** Users have reported 'permission denied' errors when trying to get `mod_proxy` (and `mod_jk`) working. Disabling SELinux (`/etc/selinux/config`) apparently fixes this.
- **Running Mac OS X:** Disable `webperfcache`, which proxies port 80 by default. A user reported this as the likely cause of Jira session problems, in the form of users' identities becoming mixed up, as below.  
 ⚠️ Additionally we do not recommend using Mac OS X as it is not supported, as in our [Supported platforms](#).

*The OSX Servers enable `webperfcache` by default for Virtual Hosts, which for static content would be great, but for dynamic instances (which ALL of ours are) it is Evil and causes many issues.*

*Of note recently was the jira session issue. Also see :-*

*<http://developer.apple.com/documentation/Darwin/Reference/ManPages/man8/webperfcache.8.html>*

*Unfortunately even if you disable `webperfcache` for a instance, if there is a single instance enabled then all instances will still proxy through `webperfcache` with resulting session problems.*

- **Too many redirects:** Both Tomcat & Apache are redirecting, when only one should be. Disable redirection in Tomcat (revert any changes as in [Running Jira over SSL or HTTPS](#)) and check that there is only one redirection in Apache.
- **General Problems:**
  1. Clear the browser cache and try again.
  2. Ensure that Jira works as expected when running directly from Tomcat and bypassing Apache. For example, accessing `http://jiraserver:8080` instead of `http://jira.atlassian.com`.
  3. Increase the **LogLevel** for Apache to debug and restart it.
  4. Attempt to access Jira and check the [Apache Log Files](#) for any errors.
  5. Raise a question on [Atlassian Answers](#) for assistance.
- **403 Forbidden error:**
  - Add the `RequestHeader unset Authorization` line to the apache configuration page to disable authorization headers.

```
<Location /jira>
  RequestHeader unset Authorization
  ProxyPreserveHost On
  ProxyPass http://jiraserver/jira
  ProxyPassReverse http://jiraserver/jira
</Location>
```

### See also

- [Integrating Jira with Apache using SSL](#)
- [Configuring Apache Reverse Proxy Using the AJP Protocol](#)

- For more advanced `mod_webapp` configurations (eg. SSL), see [this mod\\_proxy guide](#).
- [Apache Virtual Host documentation](#)

# Configuring Apache Reverse Proxy Using the AJP Protocol

 Atlassian applications allow the use of reverse-proxies within our products, however Atlassian Support does not provide assistance for configuring them. Consequently, Atlassian **can not guarantee providing any support for them.**

If assistance with configuration is required, please raise a question on [Atlassian Community](#).

## Ghostcat vulnerability in Apache Tomcat

Following this guide might make your Jira instance prone to a recent high-risk vulnerability found in Apache Tomcat.

We recommend that you wait until Jira is bundled with the Tomcat version that fixes this issue, we'll update this note once its released. For more info about this vulnerability, see:

- [CVE-2020-1938: Ghostcat - Apache Tomcat AJP File Read/Inclusion Vulnerability](#)

Until then, if you need to use the AJP Connector, there are steps you can take to mitigate this issue. For more info, see [this article](#).

This page describes how to integrate [Apache HTTP Server](#) (also referred to as `httpd`) with Jira, utilizing [mod\\_proxy\\_ajp](#) so that Apache operates as a reverse-proxy. AJP is a wire protocol and is an optimized version of the HTTP protocol to allow a standalone web server such as [Apache](#) to talk to Tomcat.

This protocol can be used in favor of `HTTP/1.1` as in either of the following Apache configurations:

- [Integrating Jira with Apache](#)
- [Integrating Jira with Apache using SSL](#)

### On this page:

- [Step 1: Configure Tomcat](#)
- [Step 2: Configure Apache HTTP Server](#)
  - [2.1 Enable the Proxy Modules](#)
  - [2.2. Configure Apache to use those Modules](#)
  - [2.3 Redirect HTTP to HTTPS](#)
- [Step 3: Configure Jira](#)
- [Troubleshooting](#)
- [See also](#)

## Step 1: Configure Tomcat

1. Stop Jira.
2. Enable the AJP Connector on the Tomcat container hosting Jira by uncommenting the following element in `$(JIRA_INSTALL)/conf/server.xml`:

```
<Connector port="8009" URIEncoding="UTF-8" enableLookups="false" protocol="AJP/1.3" />
```

3. Start Jira.

4. Test that Jira is accessible on the standard HTTP connector, for example `http://jiraserver:8080`. This is to ensure that Tomcat has successfully restarted.

## Step 2: Configure Apache HTTP Server

The installation of Apache and configuration of a DNS is not covered in this documentation. Additionally, it is assumed that Apache 2.2 has been installed and DNS entries have been configured for the Jira domain. As Apache's configuration is specific to the operation system that is used, only some distributions and their configurations are currently documented.

### 2.1 Enable the Proxy Modules

Debian/Ubuntu

1. Enable the module with the following:

```
$ sudo a2enmod proxy_ajp
Considering dependency proxy for proxy_ajp:
Module proxy already enabled
Enabling module proxy_ajp.
To activate the new configuration, you need to run:
    service apache2 restart
```

2. Restart Apache.

Windows/Other OS

1. Locate and edit the `httpd.conf` file, adding the below lines:

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
```

2. Restart Apache.

### 2.2. Configure Apache to use those Modules

Debian/Ubuntu

1. Switch into user `root`.
2. Backup the existing site or create a new one. Creating a new site is not covered within this documentation (copying the default should be sufficient).
3. Modify the existing site within `$APACHE_INSTALL/sites-available`, for example `default` (HTTP) or `default-ssl` (HTTPS).
4. Add the following inside the `VirtualHost`, replacing `jiraserver` with the hostname of the Jira server and also modifying the port if required.

```
# Jira AJP Proxy Configuration:
<Proxy *>
    Order deny,allow
    Allow from all
</Proxy>

ProxyRequests          Off
ProxyPass              /      ajp://jiraserver:8009/
ProxyPassReverse       /      ajp://jiraserver:8009/
```

5.  Missing a forward slash at the end of the URL will cause proxy errors - ensure this is in place! (Optional): Enable the site with the following:

```
# a2ensite jira
Enabling site jira.
To activate the new configuration, you need to run:
    service apache2 reload
```

**i** This is only required if a new site has been created in favor of using the default.

6. **If using HTTP, skip to step 8.** For HTTPS, the certificates need to be installed by copying the certificate and private key to the appropriate directories and the following will also need to be added to the site:

```
SSLProxyEngine          On
```

7. Include them in the Apache configuration, within the `VirtualHost` as below:

```
SSLCertificateFile      /etc/ssl/certs/jira.crt
SSLCertificateKeyFile   /etc/ssl/private/jira.key
```

8. Reload the Apache configuration.
9. Test by accessing Jira through Apache, for example <http://jira.com> or <http://atlassian.com/jira>.

#### Windows/Other OS

1. Locate and edit the `httpd.conf` file.
2. Add the following inside the `VirtualHost`, replacing `jiraserver` with the hostname of the Jira server and also modifying the port if required.

```
# Jira AJP Proxy Configuration:
<Proxy *>
    Order deny,allow
    Allow from all
</Proxy>

ProxyRequests          Off
ProxyPass               /          ajp://jiraserver:8009/
ProxyPassReverse        /          ajp://jiraserver:8009/
```

**i** Missing a forward slash at the end of the URL will cause proxy errors - ensure this is in place!

3. **If using HTTP, skip to step 5.** For HTTPS, the certificates need to be installed by copying the certificate and private key to the appropriate directories and the following will also need to be added to the site:

```
SSLProxyEngine          On
```

4. Include them in the Apache configuration, within the `VirtualHost` as below:

```
SSLCertificateFile      /etc/ssl/certs/jira.crt
SSLCertificateKeyFile   /etc/ssl/private/jira.key
```

5. Restart Apache.
6. Test by accessing Jira through Apache, for example <http://jira.com> or <http://atlassian.com/jira>.

### 2.3 Redirect HTTP to HTTPS

This is an optional step and is only required if using HTTPS. It can be done by using `mod_rewrite` (this module may require enabling), add the following to the `HTTPVirtualHost`:

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
```

### Step 3: Configure Jira

1. Set `Use gzip compression` to **OFF** as in [Configuring Jira options](#). GZIP compression is known to cause performance issues using a reverse-proxy, especially if the proxy is also compressing the traffic.

2. Set the **Base URL** to be the FQDN that Jira will be accessed on, for example <http://jira.atlassian.com>. This is also located in [Configuring Jira options](#).
  - ⚠ Jira can only be configured to respond to a single URL and the Base URL (as in [Configuring Jira options](#)) must match the URL end-users are accessing. Misconfiguration of this may cause significant problems within Jira such as the Activity Stream and Dashboard Gadgets failing to function correctly.
3. Test by accessing Jira on the FQDN (e.g.: <http://jira.atlassian.com>), ensuring that Jira is accessible and all dashboard gadgets correctly display.

## Troubleshooting

- **Hijacked Sessions:** Some users have reported problems with user sessions being hijacked when the `mod_cache` module is enabled. If these problems are encountered, try disabling the `mod_cache` module.
  - ⬆ This module is enabled by default in some Apache HTTP Server version 2 distributions.
- **Permission Denied Errors enabling `mod_proxy` (and `mod_jk`) on Linux distros that use SELinux:** Users have reported 'permission denied' errors when trying to get `mod_proxy` (and `mod_jk`) working. Disabling SELinux (`/etc/selinux/config`) apparently fixes this.
- **Running Mac OS X:** Disable `webperfcache`, which proxies port 80 by default. A user reported this as the likely cause of Jira session problems, in the form of users' identities becoming mixed up, as below.
  - ⚠ Additionally we do not recommend using Mac OS X as it is not supported, as in our [Supported platforms](#).

*The OSX Servers enable webperfcache by default for Virtual Hosts, which for static content would be great, but for dynamic instances (which ALL of ours are) it is Evil and causes many issues.*

*Of note recently was the jira session issue. Also see :-*

*<http://developer.apple.com/documentation/Darwin/Reference/ManPages/man8/webperfcache.8.html>*

*Unfortunately even if you disable webperfcache for a instance, if there is a single instance enabled then all instances will still proxy through webperfcache with resulting session problems.*

- **Too many redirects:** Both Tomcat & Apache are redirecting, when only one should be. Disable redirection in Tomcat (revert any changes as in [Running Jira over SSL or HTTPS](#)) and check that there is only one redirection in Apache.
- **General Problems:**
  1. Clear the browser cache and try again.
  2. Ensure that Jira works as expected when running directly from Tomcat and bypassing Apache. For example, accessing `http://jiraserver:8080` instead of <http://jira.atlassian.com>.
  3. Increase the **LogLevel** for Apache to debug and restart it.
  4. Attempt to access Jira and check the [Apache Log Files](#) for any errors.
  5. Raise a question on [Atlassian Answers](#) for assistance.
- **403 Forbidden error:**
  - Add the `RequestHeader unset Authorization` line to the apache configuration page to disable authorization headers.

```
<Location /jira>
  RequestHeader unset Authorization
  ProxyPreserveHost On
  ProxyPass http://jiraserver/jira
  ProxyPassReverse http://jiraserver/jira
</Location>
```

## See also

- [Integrating Jira with Apache](#)
- [Integrating Jira with Apache using SSL](#)
- [Apache Virtual Host documentation](#)

# Integrating Jira with Apache using SSL

**!** Atlassian applications allow the use of reverse-proxies within our products, however Atlassian Support does not provide assistance for configuring them. Consequently, Atlassian **can not guarantee providing any support for them.**

If assistance with configuration is required, please raise a question on [Atlassian Community](#).

## **!** Ghostcat vulnerability in Apache Tomcat

Following this guide might make your Jira instance prone to a recent high-risk vulnerability found in Apache Tomcat.

We recommend that you wait until Jira is bundled with the Tomcat version that fixes this issue, we'll update this note once its released. For more info about this vulnerability, see:

- [CVE-2020-1938: Ghostcat - Apache Tomcat AJP File Read/Inclusion Vulnerability](#)

Until then, if you need to use the AJP Connector, there are steps you can take to mitigate this issue. For more info, see [this article](#).

This page describes how to integrate [Apache HTTP Server](#) (also referred to as `httpd`) with Jira, utilizing `mod_proxy` & `mod_ssl` so that Apache operates as a reverse-proxy over HTTPS. If a HTTP configuration is required, please see our [Integrating Jira with Apache](#) documentation. Configuring Apache allows for running Jira on non-standard HTTP port (such as 8080) and users will be able to access Jira over standard HTTPS as their traffic will be routed through the proxy and encrypted outside of the network.

Apache can be configured to allow access to Jira in any of the following methods:

- Directly on its own domain: <https://atlassian.com/>
- As a subdomain of another domain: <https://jira.atlassian.com>
- It can also be accessed on a context path on either a domain or subdomain: <https://atlassian.com/jira>

This means the SSL certificate will be managed within Apache and not Tomcat, additionally the connection between Apache and Tomcat will not be encrypted. However, the connection between the browser and the outside network **will be encrypted**. This is suitable for configurations where the Jira server is within the same network as the Apache server and is illustrated below:

```
Client Browser -> HTTPS -> Apache Proxy -> HTTP -> Tomcat (JIRA)
```

### On this page:

- [Before you begin](#)
- [Step 1: Configure Tomcat](#)
- [Step 2: Configure Apache HTTP Server](#)
  - [2.1 Enable the Proxy Modules](#)
  - [2.2. Configure Apache to use those Modules](#)
  - [2.3 Redirect HTTP to HTTPS](#)
- [Step 3: Configure Jira](#)
- [Troubleshooting](#)
- [See Also](#)

This is a common configuration for networks with multiple SSL certificates and/or web applications as they are all managed in one location (Apache).

If a more complicated solution is required, refer to the [Apache HTTP Server Version Documentation](#), consult with the Apache SME within your organization, and if need be, raise a question on [Atlassian Answers](#), or get in touch with one of our [Atlassian Experts](#).

1. Jira is running on port 8080 on a server within the LAN that cannot be accessed externally (the router/firewall is not forwarding port 8080 to it).
2. Apache is set up on another server (or the same server as Jira) that can be accessed externally on HTTPS (443).

3. Apache is then accessed over HTTPS on the appropriate URL (VirtualHost), routing the traffic to and from the Jira server.

## Before you begin

⚠ It is expected that the SSL certificate has been signed by a CA and is in the PEM format prior to configuring Apache. For assistance preparing and generating SSL certificates, please consult with a SSL Vendor (for example, GoDaddy, Verisign, RapidSSL).

Identifying whether to use a domain, subdomain or context path largely depends on the type of SSL certificate provided and also any business rules around website configurations. For SSL to function without error, the domain must match the Common Name (CN) of the certificate.

This table indicates which URLs will work with the certificate CN and also makes a recommendation on the URL to use.

Jira FQDN	Common Name	Valid	Recommend Jira FQDN
<a href="https://jira.atlassian.com">https://jira.atlassian.com</a>	jira.atlassian.com	✓	<a href="https://jira.atlassian.com">https://jira.atlassian.com</a>
<a href="https://jira.atlassian.com">https://jira.atlassian.com</a>	*.atlassian.com	✓	<a href="https://jira.atlassian.com">https://jira.atlassian.com</a>
<a href="https://jira.atlassian.com">https://jira.atlassian.com</a>	atlassian.com	✗	<a href="https://atlassian.com/jira">https://atlassian.com/jira</a>
<a href="https://atlassian.com">https://atlassian.com</a>	atlassian.com	✓	<a href="https://atlassian.com/jira">https://atlassian.com/jira</a>
<a href="https://atlassian.com">https://atlassian.com</a>	jira.atlassian.com	✗	<a href="https://jira.atlassian.com">https://jira.atlassian.com</a>

A certificate that has a CN with an asterisk (\*) in it is a [wildcard certificate](#) and can support any subdomain of that domain. If you are uncertain about the URL to use, please consult with your System Administrator and the SSL vendor that provided the certificate.

## Step 1: Configure Tomcat

1. Stop Jira.
2. (Optional: If Jira does not require a context path, skip this step.)

Edit Tomcat's `server.xml` to include the required Jira context path. The below example uses `path="jira"` - this means Jira is accessible on `http://jiraserver:8080/jira` given the default Jira port is used.

```

<Engine defaultHost="localhost" name="Catalina">
  <Host appBase="webapps" autoDeploy="true" name="localhost" unpackWARs="true">
    <Context docBase="${catalina.home}/atlassian-jira" path="/jira" reloadable="
false" useHttpOnly="true">

      <!--

=====
      Note, you no longer configure your database driver or connection parameters
here.

      These are configured through the UI during application setup.

=====
      -->
    <Resource auth="Container" factory="org.objectweb.jotm.
UserTransactionFactory" jotm.timeout="60" name="UserTransaction" type="javax.transaction.
UserTransaction"/>
    <Manager pathname="" />
  </Context>
</Host>

```

ⓘ Ensure the `path` value is set with a preceding forward slash (/). For example, `path="/jira"` rather than `path="jira"`.

3. Edit Tomcat's `server.xml` to include a separate connector to proxy the requests. This requires the `scheme`, `proxyName` & `proxyPort` attributes. Replace them with the appropriate domain and port of the proxy, as in the below example:

```
<Service name="Catalina">

    <!-- Apache Proxy Connector with values for scheme, proxyName and proxyPort -->
    <Connector acceptCount="100" connectionTimeout="20000" disableUploadTimeout="true"
enableLookups="false" maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25" port="8080"
protocol="HTTP/1.1" redirectPort="8443" useBodyEncodingForURI="true"
        scheme="https" proxyName="jira.atlassian.com" proxyPort="443"/>

    <!-- Standard HTTP Connector -->
    <Connector acceptCount="100" connectionTimeout="20000" disableUploadTimeout="true"
enableLookups="false" maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25" port="8081"
protocol="HTTP/1.1" redirectPort="8443" useBodyEncodingForURI="true"/>
```

4. Disable any redirections within Tomcat to HTTPS if they have been enabled - for example the changes to `WEB-INF/web.xml` in [Running Jira applications over SSL or HTTPS](#) will cause errors when using Apache.
5. Start Jira.
6. Test that Jira is accessible on the normal connector, using a context path if applicable - for example <http://jiraserver:8081/jira>.
7. Test that the new connector is in effect by accessing Jira on the appropriate proxy connector. The behavior varies depending on the context path:
  - a. If the context path is empty or root (/), visiting Jira via the proxy connector (e.g. <http://jiraserver:8080/>) should take you to Jira with a warning:

We've detected a potential problem with JIRA's Dashboard configuration that your administrator can correct. [Hide](#)

**Dashboard Diagnostics: Mismatched URL Hostname**

JIRA is reporting that it is running on the hostname 'jira.atlassian.com', which does not match the hostname used to run these diagnostics, 'localhost'. This is known to cause JIRA to construct URLs using the incorrect hostname, which will result in errors in the dashboard, among other issues.

The most common cause of this is the use of a reverse-proxy HTTP server (often Apache or IIS) in front of the application server running JIRA. While this configuration is supported, some additional setup might be necessary in order to ensure that JIRA detects the correct hostname.

The following articles describe the issue and the steps you should take to ensure that your web server and app server are configured correctly:

- [Gadgets do not display correctly after upgrade to JIRA 4.0](#)
- [Integrating JIRA with Apache](#)
- [Integrating JIRA with Apache using SSL](#)

If you believe this diagnosis is in error, or you have any other questions, please contact [Atlassian Support](#).

**Detailed Error**  
[Click here to learn more](#)

- b. If the context path is other than empty or root(/), e.g. `/jira`, visiting Jira via the proxy connector (e.g. <http://jiraserver:8080/jira>) should redirect you to the configured proxy (e.g. <https://jira.atlassian.com/jira>).

We use two different Tomcat connectors so that testing can be done on Jira, bypassing the proxy when needed as this is a useful step when troubleshooting. It is expected that the standard connector will not be allowed external access from outside the network (the firewall will not forward any ports to it).

## Step 2: Configure Apache HTTP Server

The installation of Apache and configuration of a DNS is not covered in this documentation. Additionally, it is assumed that Apache 2.2 has been installed and DNS entries have been configured for the Jira domain. As Apache's configuration is specific to the operation system that is used, only some distributions and their configurations are currently documented.

### 2.1 Enable the Proxy Modules

Debian/Ubuntu

1. Enable the module with the following:

```
$ sudo a2enmod proxy_http ssl
Considering dependency proxy for proxy_http:
Enabling module proxy.
Enabling module proxy_http.
Enabling module ssl.
See /usr/share/doc/apache2.2-common/README.Debian.gz on how to configure SSL and create self-
signed certificates.
To activate the new configuration, you need to run:
    service apache2 restart
```

## 2. Restart Apache.

### Windows/Other OS

1. Locate and edit the `httpd.conf` file, adding the below lines if they do not already exist:

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule ssl_module modules/mod_ssl.so
```

## 2. Restart Apache.

### 2.2. Configure Apache to use those Modules

#### Debian/Ubuntu

1. Switch into user `root`.
2. Backup the existing instance or create a new one. Creating a new instance is not covered within this documentation (copying the default should be sufficient).
3. Modify the existing instance within `$APACHE_INSTALL/sites-available`, for example `default-ssl`.
4. Add the following inside the `VirtualHost`, replacing `jiraserver` with the hostname of the Jira server and also modifying the port if required.

On its own domain or subdomain:

```
# Jira Proxy Configuration:
<Proxy *>
    Order deny,allow
    Allow from all
</Proxy>

SSLProxyEngine          On
ProxyRequests           Off
ProxyPreserveHost       On
ProxyPass                /          http://jiraserver:8080/
ProxyPassReverse         /          http://jiraserver:8080/
```

**i** Missing a forward slash at the end of the URL will cause proxy errors - ensure this is in place!

Using a context path:

```
# Jira Proxy Configuration:
<Proxy *>
    Order deny,allow
    Allow from all
</Proxy>

SSLProxyEngine          On
ProxyRequests           Off
ProxyPreserveHost       On
ProxyPass                /jira          http://jiraserver:8080/jira
ProxyPassReverse         /jira          http://jiraserver:8080/jira
```

**i** The path used must be identical to the Tomcat context path. For example, forwarding `/jira` to `/jira520` cannot be done without considerable rewrite rules that are not always reliable.

5. Enable the instance with the following:

```
# a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
    service apache2 reload
```

6. Copy the certificate and private key to the appropriate directories.
7. Include them in the Apache configuration, within the `VirtualHost` as below:

```
SSLCertificateFile    /etc/ssl/certs/jira.crt
SSLCertificateKeyFile /etc/ssl/private/jira.key
```

8. (OPTIONAL): Configuration of `SSLCertificateChainFile` will contain the intermediate certificates provided by the CA vendor who signed it. Please follow consult with the CA vendor to verify if this is required.

```
SSLCertificateChainFile /etc/ssl/certs/jiraintermediate.crt
```

9. Reload the Apache configuration.
10. Test by accessing Jira through Apache, for example <http://jira.com> or <http://atlassian.com/jira>.

#### Windows/Other OS

1. Locate and edit the `httpd.conf` file.
2. Add the following inside the `VirtualHost`, replacing `jiraserver` with the hostname of the Jira server and also modifying the port if required.

#### On its own domain or subdomain:

```
# Jira Proxy Configuration:
<Proxy *>
    Order deny,allow
    Allow from all
</Proxy>

SSLProxyEngine          On
ProxyRequests           Off
ProxyPreserveHost       On
ProxyPass                /          http://jiraserver:8080/
ProxyPassReverse        /          http://jiraserver:8080/
```

**i** Missing a forward slash at the end of the URL will cause proxy errors - ensure this is in place!

#### Using a context path:

```
# Jira Proxy Configuration:
<Proxy *>
    Order deny,allow
    Allow from all
</Proxy>

SSLProxyEngine          On
ProxyRequests           Off
ProxyPreserveHost       On
ProxyPass                /jira          http://jiraserver:8080/jira
ProxyPassReverse        /jira          http://jiraserver:8080/jira
```

**i** The path used must be identical to the Tomcat context path. For example, forwarding `/jira` to `/jira520` cannot be done without considerable rewrite rules that are not always reliable.

3. Copy the certificate and private key to the appropriate directories.
4. Include them in the Apache configuration, within the `VirtualHost` as below:

```
SSLCertificateFile      /etc/ssl/certs/jira.crt
SSLCertificateKeyFile   /etc/ssl/private/jira.key
```

5. (OPTIONAL): Configuration of `SSLCertificateChainFile` will contain the intermediate certificates provided by the CA vendor who signed it. Please follow consult with the CA vendor to verify if this is required.

```
SSLCertificateChainFile /etc/ssl/certs/jiraintermediate.crt
```

6. Restart Apache.
7. Test by accessing Jira through Apache, for example <http://jira.com> or <http://atlassian.com/jira>.

### 2.3 Redirect HTTP to HTTPS

This can be done with either of the following:

- Set up the `HTTPVirtualHost` to forward to the same Tomcat Connector. Tomcat will redirect to HTTPS using the `scheme,proxyName&proxyPort` parameters. This can be done as in our [Integrating Jira with Apache](#) documentation.
- Using `mod_rewrite` (this module may require enabling), add the following to the `HTTPVirtualHost`:

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
```

### Step 3: Configure Jira

1. Set `Use gzip compression` to **OFF** as in [Configuring Jira options](#). GZIP compression is known to cause performance issues using a reverse-proxy, especially if the proxy is also compressing the traffic.
2. Set the **Base URL** to be the FQDN that Jira will be accessed on, for example <https://jira.atlassian.com>. This is also located in [Configuring Jira options](#).
  - ⚠ Jira can only be configured to respond to a single URL and the Base URL (as in [Configuring Jira options](#)) must match the URL end-users are accessing. Misconfiguration of this may cause significant problems within Jira such as the Activity Stream and Dashboard Gadgets failing to function correctly.
3. Test by accessing Jira on the FQDN (e.g.: <https://jira.atlassian.com>), ensuring that Jira is accessible and all dashboard gadgets correctly display.

### Troubleshooting

- **Hijacked Sessions:** Some users have reported problems with user sessions being hijacked when the `mod_cache` module is enabled. If these problems are encountered, try disabling the `mod_cache` module.
  - This module is enabled by default in some Apache HTTP Server version 2 distributions.
- **Permission Denied Errors enabling `mod_proxy` (and `mod_jk`) on Linux distros that use SELinux:** Users have reported 'permission denied' errors when trying to get `mod_proxy` (and `mod_jk`) working. Disabling SELinux (`/etc/selinux/config`) apparently fixes this.
- **Running Mac OS X:** Disable `webperfcache`, which proxies port 80 by default. A user reported this as the likely cause of Jira session problems, in the form of users' identities becoming mixed up, as below.
  - ⚠ Additionally we do not recommend using Mac OS X as it is not supported, as in our [Supported platforms](#).

The OSX Servers enable `webperfcache` by default for Virtual Hosts, which for static content would be great, but for dynamic instances (which ALL of ours are) it is Evil and causes many issues.

Of note recently was the `jira` session issue. Also see :-

<http://developer.apple.com/documentation/Darwin/Reference/ManPages/man8/webperfcache.8.html>

Unfortunately even if you disable `webperfcache` for a instance, if there is a single instance enabled then all instances will still proxy through `webperfcache` with resulting session problems.

- **Too many redirects:** Both Tomcat & Apache are redirecting, when only one should be. Disable redirection in Tomcat (revert any changes as in [Running Jira over SSL or HTTPS](#)) and check that there is only one redirection in Apache.
- **General Problems:**
  1. Clear the browser cache and try again.
  2. Ensure that Jira works as expected when running directly from Tomcat and bypassing Apache. For example, accessing `http://jiraserver:8080` instead of `http://jira.atlassian.com`.
  3. Increase the `LogLevel` for Apache to debug and restart it.
  4. Attempt to access Jira and check the [Apache Log Files](#) for any errors.
  5. Raise a question on [Atlassian Answers](#) for assistance.
- **403 Forbidden error:**
  - Add the `RequestHeader unset Authorization` line to the apache configuration page to disable authorization headers.

```
<Location /jira>
  RequestHeader unset Authorization
  ProxyPreserveHost On
  ProxyPass http://jiraserver/jira
  ProxyPassReverse http://jiraserver/jira
</Location>
```

## See Also

- [Integrating Jira with Apache](#)
- [Configuring Apache Reverse Proxy Using the AJP Protocol](#)
- For more advanced `mod_webapp` configurations (eg. SSL), see [this mod\\_proxy guide](#).
- [Apache Virtual Host documentation](#)

# Troubleshooting Apache

- **Hijacked Sessions:** Some users have reported problems with user sessions being hijacked when the `mod_cache` module is enabled. If these problems are encountered, try disabling the `mod_cache` module.  
📌 This module is enabled by default in some Apache HTTP Server version 2 distributions.
- **Permission Denied Errors enabling `mod_proxy` (and `mod_jk`) on Linux distros that use SELinux:** Users have reported 'permission denied' errors when trying to get `mod_proxy` (and `mod_jk`) working. Disabling SELinux (`/etc/selinux/config`) apparently fixes this.
- **Running Mac OS X:** Disable `webperfcache`, which proxies port 80 by default. A user reported this as the likely cause of Jira session problems, in the form of users' identities becoming mixed up, as below.  
⚠️ Additionally we do not recommend using Max OS X as it is not supported, as in our [Supported platforms](#).

*The OSX Servers enable `webperfcache` by default for Virtual Hosts, which for static content would be great, but for dynamic instances (which ALL of ours are) it is Evil and causes many issues.*

*Of note recently was the jira session issue. Also see :-*

*<http://developer.apple.com/documentation/Darwin/Reference/ManPages/man8/webperfcache.8.html>*

*Unfortunately even if you disable `webperfcache` for a instance, if there is a single instance enabled then all instances will still proxy through `webperfcache` with resulting session problems.*

- **Too many redirects:** Both Tomcat & Apache are redirecting, when only one should be. Disable redirection in Tomcat (revert any changes as in [Running Jira over SSL or HTTPS](#)) and check that there is only one redirection in Apache.
- **General Problems:**
  1. Clear the browser cache and try again.
  2. Ensure that Jira works as expected when running directly from Tomcat and bypassing Apache. For example, accessing `http://jiraserver:8080` instead of `http://jira.atlassian.com`.
  3. Increase the `LogLevel` for Apache to debug and restart it.
  4. Attempt to access Jira and check the [Apache Log Files](#) for any errors.
  5. Raise a question on [Atlassian Answers](#) for assistance.
- **403 Forbidden error:**
  - Add the `RequestHeader unset Authorization` line to the apache configuration page to disable authorization headers.

```
<Location /jira>
  RequestHeader unset Authorization
  ProxyPreserveHost On
  ProxyPass http://jiraserver/jira
  ProxyPassReverse http://jiraserver/jira
</Location>
```

# Securing Jira applications with Apache HTTP Server

The following outlines some basic techniques to secure a Jira instance using Apache HTTP Server. These instructions are basic to-do lists and should not be considered comprehensive. For more advanced security topics see the "Further Information" section below.

- [Using Apache to limit access to the Jira administration interface](#)
- [Using Fail2Ban to limit login attempts](#) (Jira 4.1 has login-rate limiting, but Fail2Ban can be useful for older versions and more advanced security setups.)

## Further information

- [Integrating Jira with Apache](#)

# Using Apache to limit access to the Jira administration interface

## Limiting administration to specific IP addresses

The Jira administration interface is a critical part of the application; anyone with access to it can potentially compromise not only the Jira instance but the entire machine. As well as limiting access to users who really need it, and using strong passwords, you should consider limiting access to it to certain machines on the network or internet. If you are using an [Apache HTTP Server](#), this can be done with Apache's **Location** functionality as follows.

### 1. Create a file that defines permission settings

This file can be in the Apache configuration directory or in a system-wide directory. For this example we'll call it "sysadmin\_ips\_only.conf". This file should contain the following:

```
Order Deny,Allow
Deny from All

# Mark the Sysadmin's workstation
Allow from 192.168.12.42
```

### 2. Add the file to your Virtual Host

In your Apache Virtual Host, add the following lines to restrict the administration actions to the Systems Administrator:

```
<LocationMatch Administrators.jspa>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteAttachment>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AcknowledgeTask>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ActivateWorkflow>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ActivateWorkflowStep2>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddIssueSecurity>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddIssueSecurityScheme>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddLevel>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddNotification>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddNotificationScheme>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddPermission>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddPermissionScheme>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddPopMailServer>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddProject>
  Include sysadmin_ips_only.conf
</LocationMatch>
```

```

<LocationMatch AddProjectCategory>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddRepository>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddSmtplibMailServer>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddUser>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddWorkflowScheme>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddWorkflowSchemeEntity>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddWorkflowTransition>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddWorkflowTransitionCondition>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddWorkflowTransitionConditionParams>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddWorkflowTransitionFunctionParams>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddWorkflowTransitionPostFunction>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddWorkflowTransitionValidator>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AddWorkflowTransitionValidatorParams>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AssociateFieldToScreens>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AssociateIssueTypeSchemes>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch AssociateIssueTypeSchemesWithDefault>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch BugzillaImport>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch BulkEditUserGroups>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch CloneWorkflow>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ConfigureCache>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ConfigureCsvMapping>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ConfigureCustomField>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ConfigureFieldLayout>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ConfigureFieldLayoutScheme>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ConfigureFieldScreen>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ConfigureFieldScreenScheme>

```

```

    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ConfigureFogBugzMapping>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ConfigureIssueTypeScreenScheme>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ConfigureLogging>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ConfigureOptionSchemes>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch CopyFieldLayout>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch CopyFieldLayoutScheme>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch CopyIssueSecurityScheme>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch CopyNotificationScheme>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch CopyPermissionScheme>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch CopyWorkflowScheme>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch CreateCustomField>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch CreateDraftWorkflow>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch CsvImporter>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch CurrentUsersList>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteCustomField>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteGroup>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteIssueSecurity>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteIssueSecurityLevel>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteIssueSecurityScheme>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteIssueType>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteLinkType>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteMailServer>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteNotification>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteNotificationScheme>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteOptionScheme>
    Include sysadmin_ips_only.conf

```

```

</LocationMatch>
<LocationMatch DeletePermission>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeletePermissionScheme>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeletePriority>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteProject>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteProjectCategory>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteProjectRole>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteRepository>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteResolution>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteStatus>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteSubTaskIssueType>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteTrustedApplication>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteUser>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteUserProperty>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteWorkflowScheme>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteWorkflowSchemeEntity>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteWorkflowStep>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteWorkflowTransitionCondition>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteWorkflowTransitionPostFunction>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteWorkflowTransitions>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DeleteWorkflowTransitionValidator>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch DisableSubTasks>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditAnnouncementBanner>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditApplicationProperties>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditAttachmentSettings>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditBasicConfig>
  Include sysadmin_ips_only.conf
</LocationMatch>

```

```
<LocationMatch EditCustomField>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditCustomFieldDefaults>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditCustomFieldOptions>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditDefaultFieldLayoutItem>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditFieldLayout>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditFieldLayoutItem>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditFieldLayoutItemRenderer>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditFieldLayoutItemRendererConfirmation>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditFieldLayoutScheme>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditFieldScreen>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditFieldScreenScheme>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditFieldScreenSchemeItem>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditIssueSecurities>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditIssueSecurityScheme>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditIssueType>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditIssueTypeScreenScheme>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditLinkType>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditListener>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditLookAndFeel>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditNotifications>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditNotificationScheme>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditPermissions>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditPermissionScheme>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditPriority>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditProjectCategory>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditProjectRole>
```

```

    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditResolution>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditService>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditStatus>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditSubTaskIssueTypes>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditTrustedApplication>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditUser>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditUserDefaultSettings>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditUserGroups>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditUserProjectRoles>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditUserProperties>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditUserProperty>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditWorkflow>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditWorkflowScheme>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditWorkflowSchemeEntities>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditWorkflowStep>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditWorkflowTransition>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditWorkflowTransitionConditionParams>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditWorkflowTransitionPostFunctionParams>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EditWorkflowTransitionValidatorParams>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch EnterpriseSelectProjectRepository>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ExternalImport>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch FogBugzImport>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch GlobalPermissions>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch GroupBrowser>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ImportWorkflowFromXml>
    Include sysadmin_ips_only.conf

```

```

</LocationMatch>
<LocationMatch IndexAdmin>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch IndexOptimize>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch IntegrityChecker>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch JellyRunner>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch JiraSupportRequest>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch LDAPConfigurer>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ListEventTypes>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ListWorkflows>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch MailQueueAdmin>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch MakeDefaultLevel>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ManageConfiguration>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ManageConfigurationScheme>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ManageIssueTypeSchemes>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ManageSubTasks>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch MantisImport>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch MigrateIssueTypes>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ProjectEmail>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ProjectImportBackupOverviewProgress>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ProjectImportMappingProgress>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ProjectImportMissingMandatoryUsersCannotCreate>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ProjectImportMissingMandatoryUsersExtMgmt>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ProjectImportMissingOptionalUsersCannotCreate>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ProjectImportMissingOptionalUsersExtMgmt>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ProjectImportMissingUsersAutoCreate>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ProjectImportProgress>
  Include sysadmin_ips_only.conf
</LocationMatch>

```

```

<LocationMatch ProjectImportResults>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ProjectImportSelectBackup>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ProjectImportSelectProject>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ProjectImportSummary>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch PublishDraftWorkflow>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch RepositoryTest>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ResetFailedLoginCount>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SchedulerAdmin>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SchemeComparisonPicker>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SchemeComparisonTool>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SchemeGroupToRoleMapper>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SchemeGroupToRoleResult>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SchemeGroupToRoleTransformer>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SchemeMerge>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SchemeMergePreview>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SchemeMergeResult>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SchemePicker>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SchemePurgeToolPreview>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SchemePurgeToolResults>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SchemePurgeTypePicker>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SchemeTools>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SchemeTypePicker>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SelectFieldLayoutScheme>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SelectIssueTypeSchemeForProject>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SelectIssueTypeScreenScheme>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SelectProjectCategory>

```

```
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SelectProjectIssueSecurityScheme>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SelectProjectPermissionScheme>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SelectProjectRepository>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SelectProjectScheme>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SelectProjectSecuritySchemeStep2>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SelectProjectWorkflowScheme>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SelectProjectWorkflowSchemeStep2>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SelectProjectWorkflowSchemeStep3>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SelectScreenScheme>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SendBulkMail>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SendTestMail>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ServiceExecutor>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SetGlobalEmailPreference>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch SetPassword>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch TaskAdmin>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch TimeTrackingAdmin>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch TrackbackAdmin>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch UpdatePopMailServer>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch UpdateRepository>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch UpdateSmtpMailServer>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch UserBrowser>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewApplicationProperties>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewAttachmentSettings>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewCustomFields>
    Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewDefaultProjectRoleActors>
    Include sysadmin_ips_only.conf
```

```

</LocationMatch>
<LocationMatch ViewFieldLayouts>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewFieldLayoutSchemes>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewFieldScreens>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewFieldScreenSchemes>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewGroup>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewIssueColumns>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewIssueFields>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewIssueSecuritySchemes>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewIssueTypes>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewIssueTypeScreenSchemes>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewLicense>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewLinkTypes>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewListeners>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewLogging>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewLookAndFeel>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewMemoryInfo>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewNotificationSchemes>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewPermissionSchemes>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewPlugins>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewPriorities>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewProjectCategories>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewProjectRoles>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewProjectRoleUsage>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewResolutions>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewServices>
  Include sysadmin_ips_only.conf
</LocationMatch>

```

```
<LocationMatch ViewStatuses>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewSystemInfo>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewTranslations>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewTrustedApplications>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewUpgradeHistory>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewUser>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewUserDefaultSettings>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewUserProjectRoles>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewWorkflowSchemes>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewWorkflowStep>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewWorkflowStepMetaAttributes>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewWorkflowSteps>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewWorkflowTransition>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewWorkflowTransitionConditionalResult>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewWorkflowTransitionMetaAttributes>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch ViewWorkflowXml>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch XmlBackup>
  Include sysadmin_ips_only.conf
</LocationMatch>
<LocationMatch XmlRestore>
  Include sysadmin_ips_only.conf
</LocationMatch>
```

# Using Fail2Ban to limit login attempts

 JIRA 4.1 includes a [rate-limiting mechanism](#), but older versions and other applications such as Confluence need external help from a tool such as Fail2Ban.

## What is Fail2Ban?

We need a means of defending sites against brute-force login attempts. [Fail2Ban](#) is a Python application which trails logfiles, looks for [regular expressions](#) and works with Shorewall (or directly with iptables) to apply temporary blacklists against addresses that match a pattern too often. This can be used to limit the rate at which a given machine hits login URLs for Confluence.

## Prerequisites

- Requires [Python](#) 2.4 or higher to be installed
- Requires Apache Reverse Proxy to be installed
- Needs a specific file to follow, which means your Apache instance needs to log your Confluence access to a known logfile. You **should adjust the configuration below** appropriately.

## How to set it up

*This list is a skeletal version of the instructions*

- There's an RPM available for RHEL on the [download page](#), but you can also download the source and set it up manually
- Its configuration files go into `/etc/fail2ban`
- The generic, default configuration goes into `.conf` files (`fail2ban.conf` and `jail.conf`). Don't change these, as it makes upgrading difficult.
- Overrides to the generic configuration go into `.local` files corresponding to the `.conf` files. These only need to contain the specific settings you want overridden, which helps maintainability.
- Filters go into `filter.d` this is where you define regexps, each going into its own file
- Actions go into `action.d` you probably won't need to add one, but it's handy to know what's available
- "jails" are a configuration unit that specify one regexp to check, and one or more actions to trigger when the threshold is reached, plus the threshold settings (e.g. more than 3 matches in 60 seconds causes that address to be blocked for 600 seconds)
- Jails are defined in `jail.conf` and `jail.local`. Don't forget the `enabled` setting for each one it can be as bad to have the wrong ones enabled as to have the right ones disabled.

## Running Fail2Ban

- Use `/etc/init.d/fail2ban {start|stop|status}` for the obvious operations
- Use `fail2ban-client -d` to get it to dump its current configuration to STDOUT. Very useful for troubleshooting.
- Mind the CPU usage; it can soak up resources pretty quickly on a busy site, even with simple regexp
- It can log either to syslog or a file, whichever suits your needs better

## Common Configuration

### `jail.local`

```
# The DEFAULT allows a global definition of the options. They can be override
# in each jail afterwards.

[DEFAULT]

# "ignoreip" can be an IP address, a CIDR mask or a DNS host. Fail2ban will not
# ban a host which matches an address in this list. Several addresses can be
# defined using space separator.
# ignoreip = <space-separated list of IPs>

# "bantime" is the number of seconds that a host is banned.
bantime = 600

# A host is banned if it has generated "maxretry" during the last "findtime"
# seconds.
findtime = 60

# "maxretry" is the number of failures before a host get banned.
maxretry = 3

[ssh-iptables]

enabled = false

[apache-shorewall]

enabled = true
filter = cac-login
action = shorewall
logpath = /var/log/httpd/confluence-access.log
bantime = 600
maxretry = 3
findtime = 60
backend = polling
```

## Configuring for Confluence

 The following is an example only, and you should adjust it for your site.

### filter.d/confluence-login.conf

```
[Definition]

failregex = <HOST>.*"GET /login.action

ignoreregex =
```

## Configuring for Jira

 The following is an example only, and you should adjust it for your site.

### filter.d/jira-login.conf

```
[Definition]

failregex = <HOST>.*"GET /login.jsp

ignoreregex =
```

# Changing Jira application TCP ports

## Why change Jira application TCP ports?

By default, Jira applications use TCP listening port **8080** and hence, Jira applications are typically available at `http://<yourserver>:8080`.

If, however, an existing service running on your machine is claiming port **8080**, there will be a conflict and Jira applications will fail to start. You may see errors like this:

```
LifecycleException: Protocol handler initialization failed: java.net.BindException: Address already in use: 8080
```

This can be fixed by changing Jira applications to use another TCP listening port (eg. **8100**) and shutdown port (eg. **8015**).

## Changing Jira application TCP ports

Before you change Jira application TCP ports, read the following:

- **Which port number should I choose?** If you are not sure which port number to choose, use a tool such as *netstat* to determine which port numbers are free to use by Jira applications. The highest port number that can be used is 65535 because it is the highest number which can be represented by an unsigned 16 bit binary number. [The Internet Assigned Numbers Authority \(IANA\)](#) lists the registration of commonly used port numbers for well-known Internet services, it's advisable to avoid any of those ports.
- **A note about firewalls:** When you choose a port number for Jira, bear in mind that your firewall may prevent people from connecting to Jira based on the port number. Organizations with a local network protected by a firewall typically need to consider modifying their firewall configuration whenever they install a web-based application (such as Jira) that is running on a new port or host. Even personal laptop and desktop machines often come with firewall software installed that necessitates the same sort of change as described above. If Jira does not need to be accessed from outside the firewall, then no firewall configuration changes will be necessary.

You can change Jira's TCP ports by using the **Jira configuration tool** or by **manually editing the server.xml file**. If you installed Jira using the 'Windows Installer', 'Linux Installer', or from an 'Archive File', you can use the Jira configuration tool.

### Changing Jira's TCP ports using the Jira configuration tool

1. Start the Jira configuration tool. See [Using the Jira configuration tool](#) for instructions on where to find the tool.
2. Click the **Web Server** tab.
3. In the **HTTP Port** field, enter the new TCP listening port number.
4. In the **Control Port** field, enter the new TCP shutdown port number.
5. Click the **Save** button. Your changes are saved to the `server.xml` file located in the `conf` subdirectory of your [Jira application installation directory](#).

### Changing Jira's TCP ports by editing the server.xml file

Edit the `server.xml` file in the `conf` subdirectory of the Jira installation directory. The start of the file looks like:

```
<Server port="8005" shutdown="SHUTDOWN">

  <Service name="Catalina">

    <Connector port="8080"
      maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
      enableLookups="false" redirectPort="8443" relaxedPathChars="[]" relaxedQueryChars="[]|{}^\\`&quot;&lt;
      &gt;" acceptCount="100" connectionTimeout="20000" disableUploadTimeout="true" />

    ...
```

For example, change the shutdown port from "**8005**" to "**8015**" and the listening port (i.e. in the `<connector/>` element) from "**8080**" to "**8100**". (See [below](#) to decide which TCP port numbers should be used for Jira.)

Then, restart Jira and point a browser to `http://<yourserver>:8100`

⊖ If you are running on a Unix server and bind the ports below 1024 (such as port 80 for example), you will **need to start Jira as root** in order to successfully bind to the port.

Related topics

[Changing Confluence's listening ports](#)

# Connecting to SSL services

 Atlassian applications allow the use of SSL within our applications, however Atlassian Support does not provide assistance for configuring it. Consequently, Atlassian **can not guarantee providing any support for it.**

- If assistance with conversions of certificates is required, please consult with the vendor who provided the certificate.
- If assistance with configuration is required, please raise a question on [Atlassian Answers](#).

This page describes how to get web applications like Jira and Confluence connecting to external servers over SSL, via the various SSL-wrapped protocols. For instance, you may want to:

- Refer to an `https://...` URL in a Confluence macro.
- Use an IMAPS server to retrieve mail in Jira.
- Use SMTP over SSL (SMTPS) to send mail in Jira.
- Connect to a LDAP directory over SSL.
- Set up **Trusted Applications** over SSL.

If you want to run Jira *itself* over SSL, see [Running Jira applications over SSL or HTTPS](#) or [Integrating Jira with Apache using SSL](#).

## Add SSL Certificates automatically!

We now have a Jira SSL [Atlassian Labs app](#) for this process. Please install and use the app before going through these docs.

### On this page:

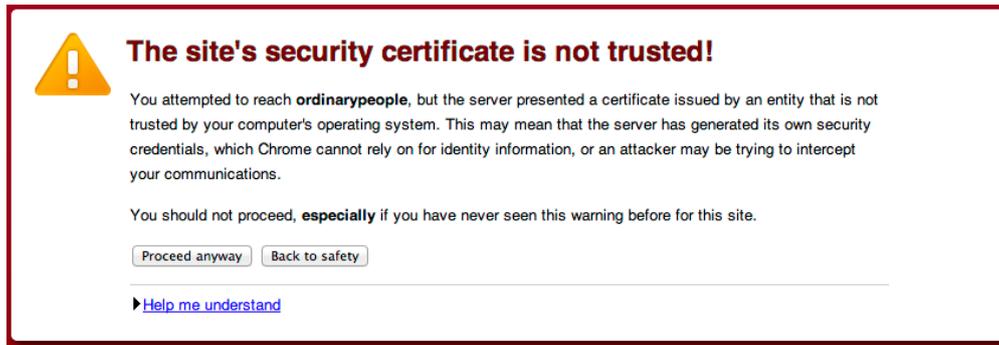
- [Problem Symptoms](#)
- [The Cause](#)
- [Resolution](#)
  - [Obtain and Import the Server's Public Certificate](#)
  - [Alternative KeyStore Locations](#)
  - [Debugging](#)

## Problem Symptoms

Attempting to access URLs that are encrypted with SSL (for example HTTPS, LDAPS, IMAPS) throws an exception and Jira refuses to connect to it. For example:

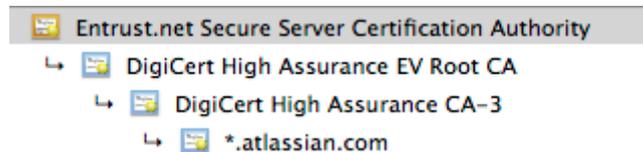
```
javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to
requested target
    at com.sun.mail.imap.IMAPStore.protocolConnect(IMAPStore.java:441)
    at javax.mail.Service.connect(Service.java:233)
    at javax.mail.Service.connect(Service.java:134)
```

This is the same as the following error that's generated in Chrome when visiting a page that's encrypted with a self-signed certificate, except Java can't "Proceed anyway", it just refuses the certificate:



## The Cause

Whenever Jira attempts to connect to another application over SSL (e.g.: HTTPS, IMAPS, LDAPS), it will *only* be able to connect to that application if it can trust it. The way trust is handled in the Java world (this is what Jira is written in) is that you have a keystore (typically `$JAVA_HOME/lib/security/cacerts`) or also known as the trust store. This contains a list of all the known CA certificates and Java will only trust certificates that are signed by those CA certificate or public certificates that exist within that keystore. For example, if we look at the certificate for Atlassian:



We can see the `*.atlassian.com` certificate has been signed by the intermediate certificates, **DigiCert High Assurance EV Root CA** and **DigiCert High Assurance CA-3**. These intermediate certificates have been signed by the root **Entrust.net Secure Server CA**. Those three certificates combined are referred to as the certificate chain. As all of those CA certificates are within the Java keystore (`cacerts`), Java will trust any certificates signed by them (in this case, `*.atlassian.com`). Alternatively, if the `*.atlassian.com` certificate was in the keystore, Java would also trust that site.

This problem comes from a certificate that is either self-signed (a CA did not sign it) or the certificate chain does not exist within the Java keystore. Subsequently, Jira doesn't trust the certificate and fails to connect to the application.

## Resolution

In order to resolve this, the public certificate need to be imported in the Java keystore that Jira uses. In the example above, this is `*.atlassian.com` and we cover how to install it below.

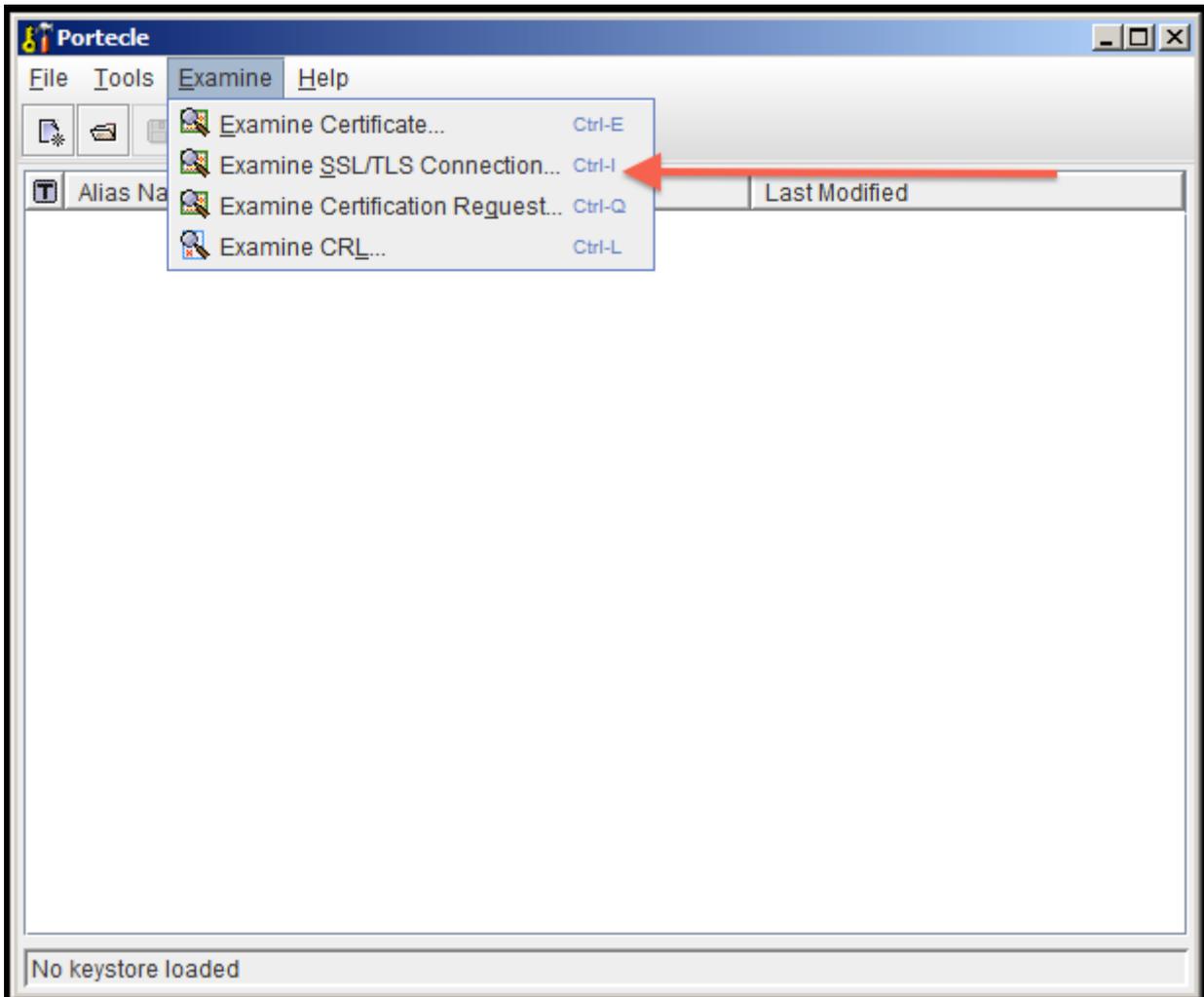
**i** If you're unable to install Portecle on the server or prefer the command line please see our [Command Line Installation](#) section below.

### Obtain and Import the Server's Public Certificate

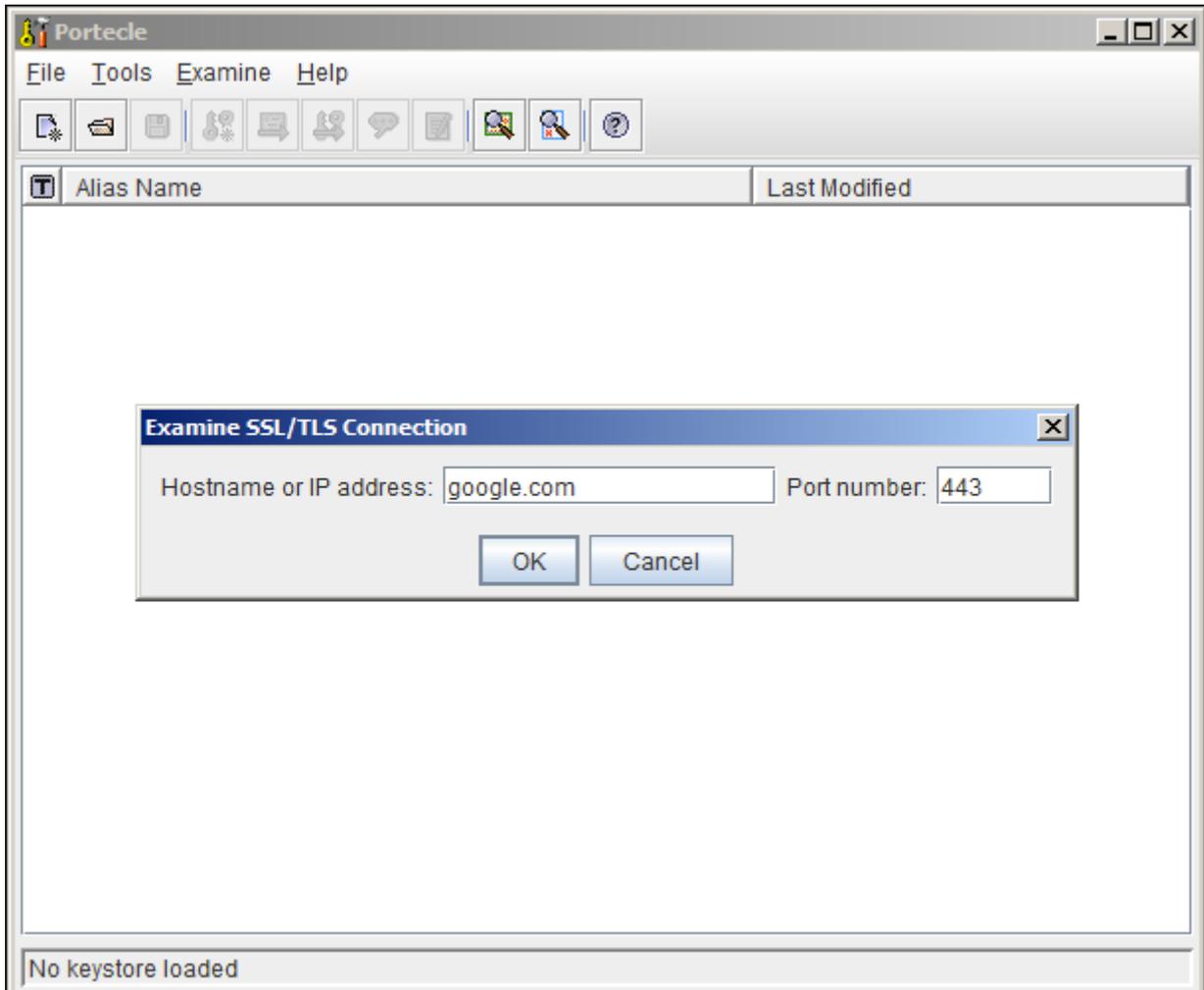
1. Download and install the [Portecle](#) app onto the server that runs Jira.
  - ⚠ This is a third-party application and not supported by Atlassian.
2. Ensure the `<JAVA_HOME>` variable is pointing to the same version of Java that Jira uses. See our [Setting JAVA\\_HOME](#) docs for further information on this.
  - 📘 If running on a Linux/UNIX server, X11 will need to be forwarded when connecting to the server (so you can use the GUI), as below:

```
ssh -X user@server
```

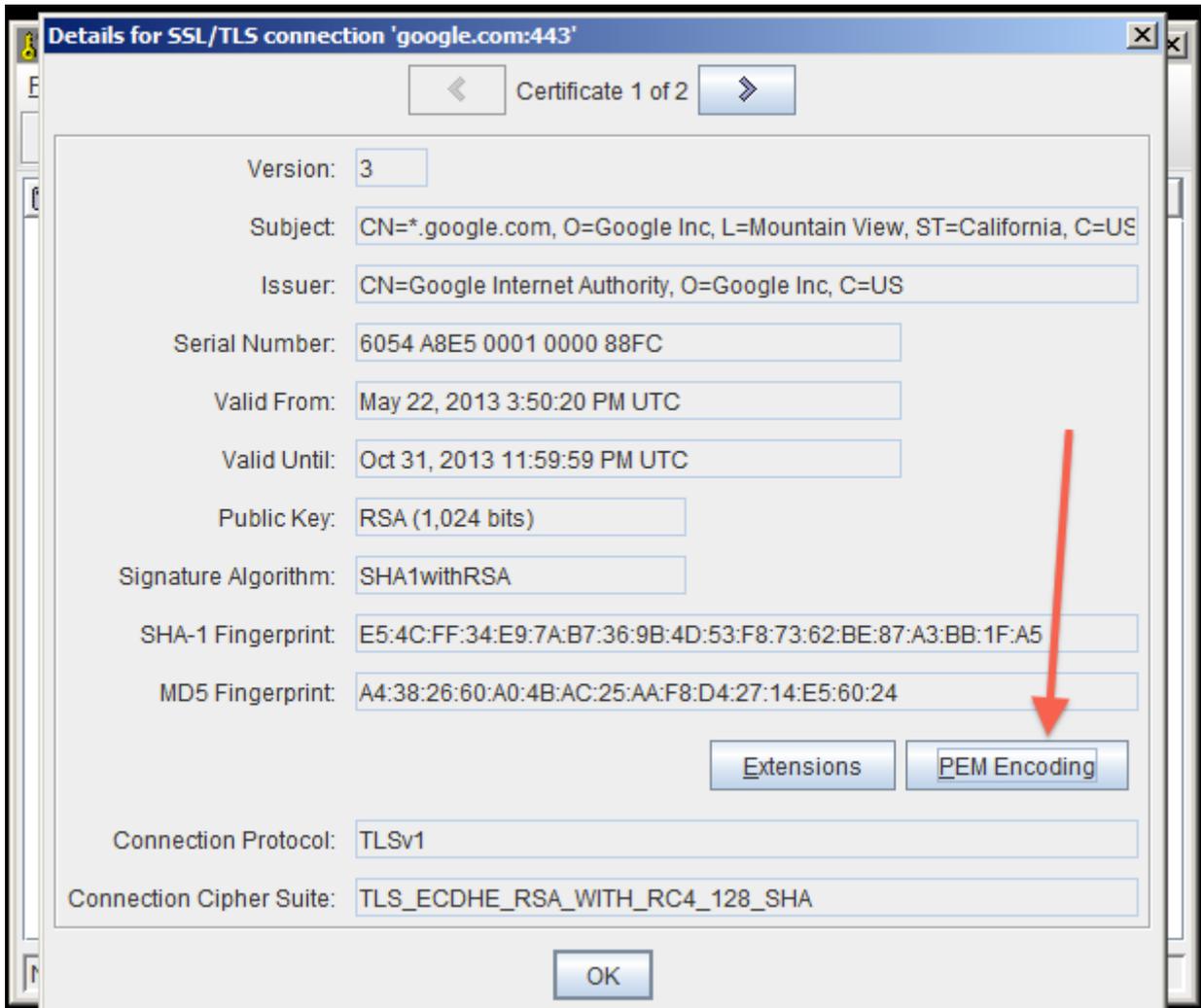
3. Select the **Examine** menu and then click **Examine SSL/TLS Connection**:



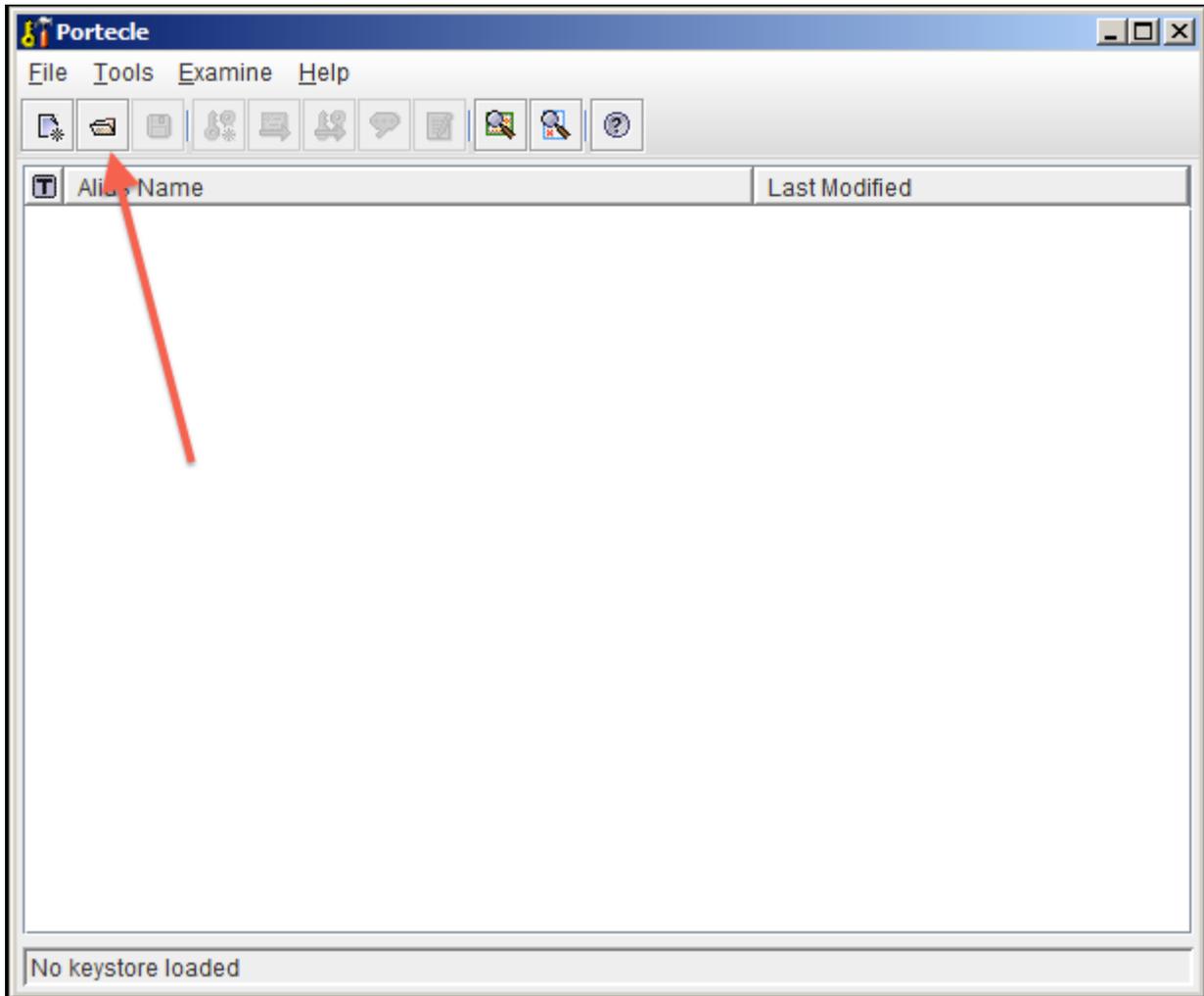
4. Enter the SSL Host and Port of the target system:



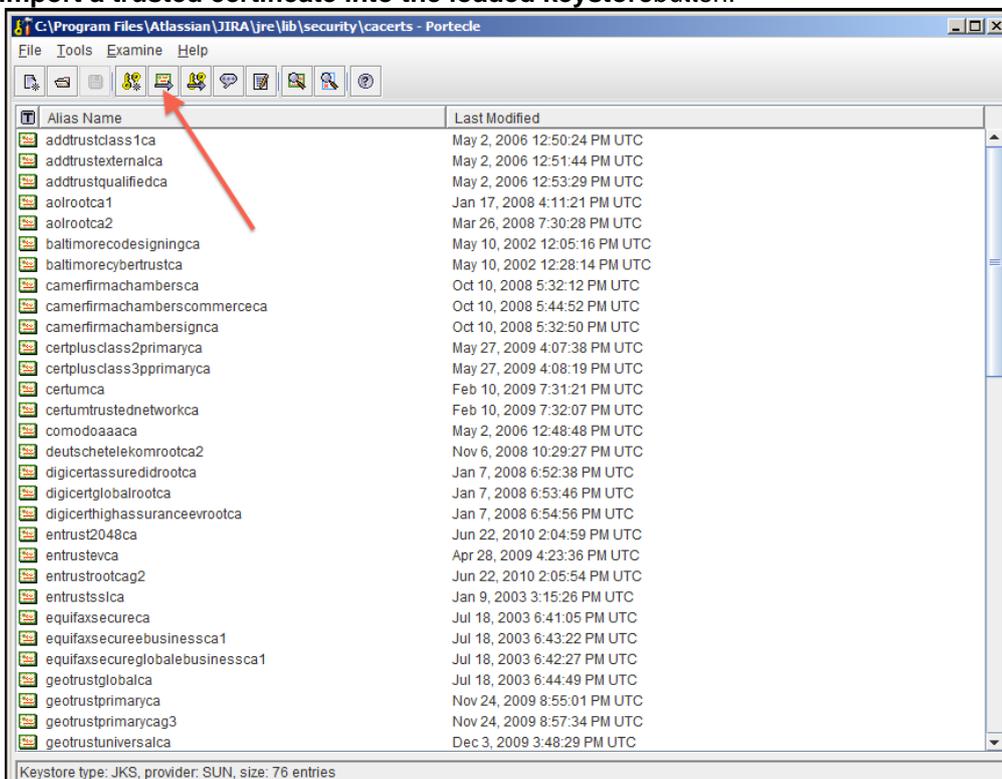
5. Wait for it to load, then select the public certificate and click on PEM:



6. Export the certificate and save it.
7. Go back to the main screen and select the **Open an existing keystore from disk** option, select `cacerts` (for example `$JAVA_HOME/lib/security/cacerts`) then enter the password (the default is `changeit`).

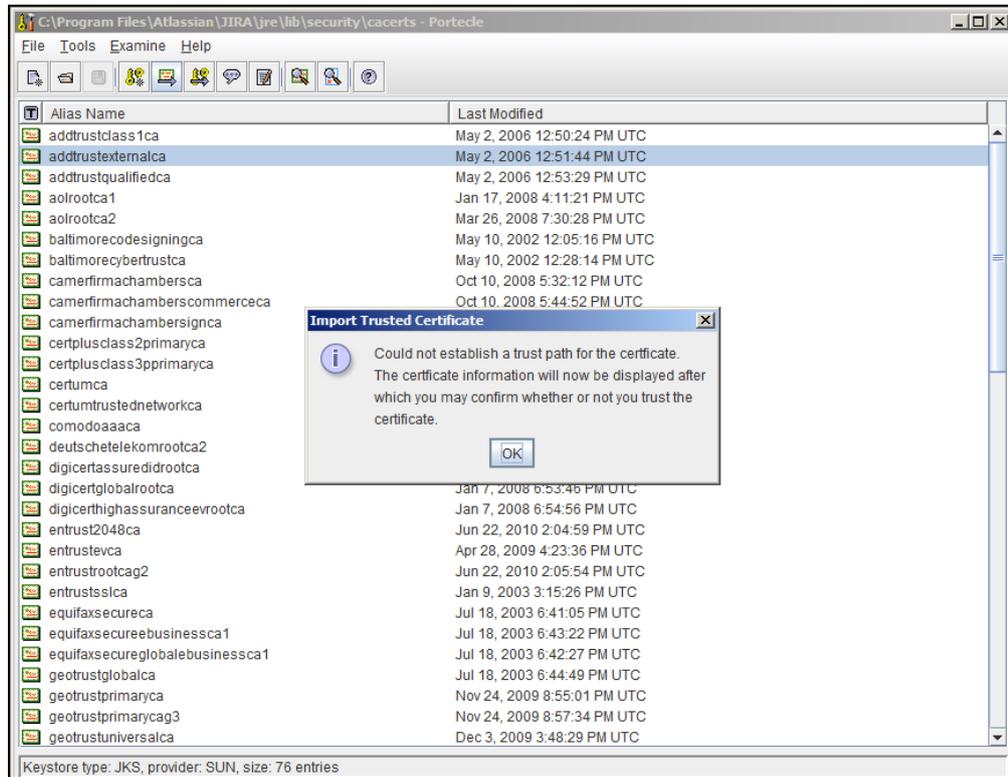


8. Select the **Import a trusted certificate into the loaded keystore** button:



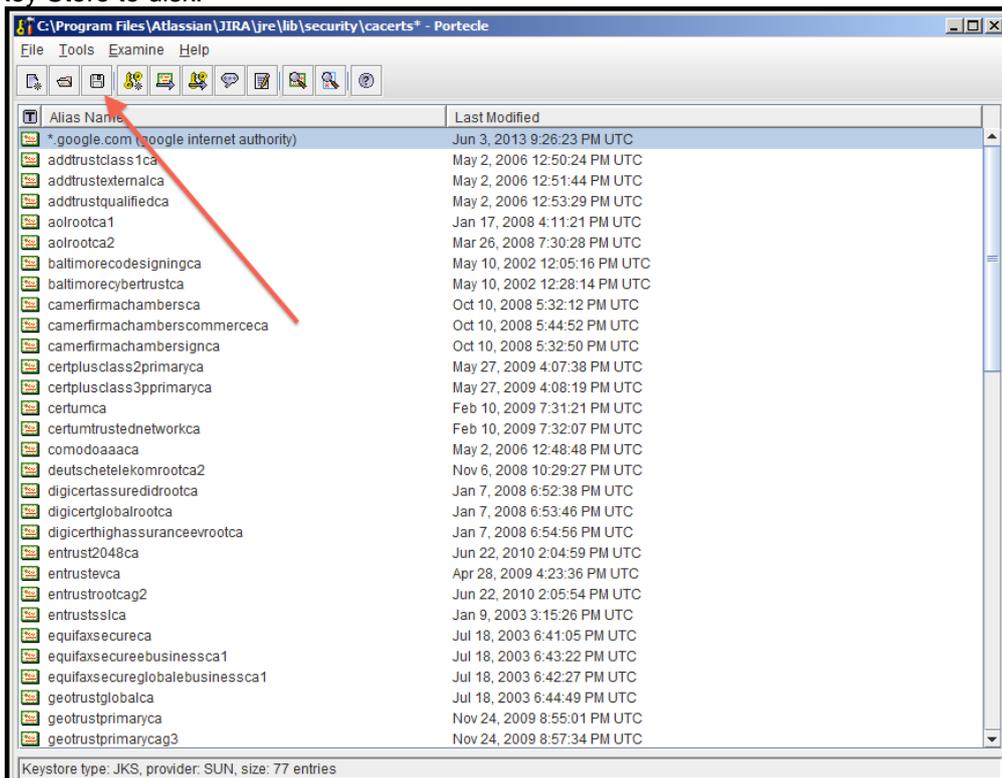
9. Select the certificate that was saved in step 6 and confirm that you trust it, giving it an appropriate alias (e.g.: confluence).

- You may hit this error:



b. If so, hit OK, and then accept the certificate as trusted.

10. Save the Key Store to disk:



11. Restart Jira.

12. Test that you can connect to the host.

## Command Line Installation

1. Fetch the certificate, replacing [google.com](#) with the FQDN of the server Jira is attempting to connect to:  
**Unix:**

```
openssl s_client -connect google.com:443 < /dev/null | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > public.crt
```

**Windows:**

```
openssl s_client -connect google.com:443 < NUL | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > public.crt
```

**i** The command above will only be executed if you have [Sed for Windows](#) as well as [OpenSSL](#) installed on your environment. If you don't have Sed you don't want to install it, use the instructions below as an alternative. Issue the following command:

```
openssl s_client -connect google.com:443
```

Save the output to a file called `public.crt`. Edit the `public.crt` file so it contains only what is between the `BEGIN CERTIFICATE` and `END CERTIFICATE` lines. This is how your file should look like after you edited it:

```
-----BEGIN CERTIFICATE-----
< Certificate content as fetched by the command line.
Don't change this content, only remove what is before
and after the BEGIN CERTIFICATE and END CERTIFICATE.
That's what your Sed command is doing for you :-) >
-----END CERTIFICATE-----
```

2. Import the certificate:

```
<JAVA_HOME>/keytool -import -alias <server_name> -keystore <JAVA_HOME>/lib/security/cacerts -file public.crt
```

If you are using Windows, make sure to quote the entire cacerts path.

```
<JAVA_HOME>\keytool -import -alias <server_name> -keystore "F:\Program Files\Atlassian\JIRA\jre\lib\security\cacerts" -file public.crt
```

**Alternative KeyStore Locations**

Java will normally use a system-wide keystore in `$JAVA_HOME/jre/lib/security/cacerts`, but it is possible to use a different keystore by specifying a parameter, **-Djavax.net.ssl.trustStore=/path/to/keystore**, where `'/path/to/keystore'` is the absolute file path of the alternative keystore.

However, setting this is **not recommended** because if Java is told to use a custom keystore (eg. containing a self-signed certificate), then Java will not have access to the root certificates of signing authorities found in `$JAVA_HOME/jre/lib/security/cacerts`, and accessing most CA-signed SSL sites will fail. It is better to add new certificates (eg. self-signed) to the system-wide keystore (as above).

**Debugging**

Problems are typically one of two forms:

- The certificate was installed into the incorrect keystore.
- The keystore does not contain the certificate of the SSL service you're connecting to.

See Also

- [Configuring an SSL Connection to Active Directory](#)
- [Running Jira applications over SSL or HTTPS](#)
- [Integrating Jira with Apache using SSL](#)

# Running Jira applications over SSL or HTTPS

 You can use SSL with Atlassian applications. However, SSL configuration is outside the scope of Atlassian Support. Still, here's what we can recommend:

- If you need help with converting certificates, consult with the vendor who provided the certificate.
- If you need help with configuring SSL, create a question on [the Atlassian Community](#).

Be aware that SHA-1 is being phased out due to known weaknesses.

This article describes how to run Jira applications over SSL or HTTPS by configuring Apache Tomcat with HTTPS. This procedure only covers the common installation types of Jira. It's not definitive or comprehensive guide to configuring HTTPS and may not apply to your environment.

You can also find more information on the subject in the following articles:

- [How to run JIRA over HTTPS with a Personal Information Exchange \(PEX\) Certificate](#)
- [Integrating Jira with Apache using SSL](#)
- [How to import a public SSL certificate into a JVM](#)

**Why should you run Jira over SSL or HTTPS?** When people access web applications, there is always a possibility that their usernames and passwords can be intercepted by intermediaries between your computer and the ISP (Internet Service Provider)/company. It's a good idea to enable access via HTTPS (HTTP over SSL) and make this a requirement for pages where passwords are sent. Note, however, that using HTTPS may result in slower performance.

 Running Jira without HTTPS enabled may leave your instance exposed to vulnerabilities, such as Man in the middle or DNS Rebinding attacks. We recommend that you enable HTTPS on your instance.

## Before you begin

### Support

Atlassian Support will refer SSL support to the Certificate Authority (CA) that issues the Certificate. The SSL-related instructions on this page are provided as a reference only.

### Windows installers

The [Windows Installer](#) installs its own Java Runtime Environment (JRE) Java platform, which is used to run Tomcat. When updating SSL certificates, please do so in this JRE installation.

### Related bugs

Jira 7.3 and later is affected by two bugs that incorrectly set the protocol in the `server.xml` file. You can work around this issue by setting the protocol manually.

Two bugs affecting Jira 7.3.0 and later:

 [JRASERVER-63734](#) - Tomcat fails to start in 7.3+ due to protocol deprecation in Tomcat 8.5  
**CLOSED**

 [JRASERVER-64082](#) - Jira Configuration Tool sets wrong value of the 'protocol' attribute for Tomcat SSL configuration **CLOSED**

The workaround is to manually edit the `server.xml` file to change the protocol on your HTTPS connector to:

```
protocol="org.apache.coyote.http11.Http11NioProtocol"
```

## Jira behind a reverse-proxy

If hosting Jira behind a reverse-proxy, such as Apache, see [Integrating Jira with Apache using SSL](#) for more information.

## Adding new connections

When you add a new connection, like an SSL one, the Jira config tool saves an entry with connection details in the server.xml file. This entry doesn't include properties that handle special characters, so you'll need to add them manually. This is required, as Jira won't work properly without it. We've described the required steps below, but you can read more about it [here](#).

## Insecure BKS-V1 keystore format



Due to a security vulnerability of the BKS-V1 keystore format (provided by the BouncyCastle library), we recommend that you don't use it in your Jira instance. [Learn more](#)

## Generate the Java KeyStore

In this section, you will create a Java Key Store (JKS), which will hold your SSL certificates. The SSL certificates are required for SSL to work in Jira. In the SSL world, certificates fall into two major categories:

Certificate	Description	When to use	Steps
Self-signed	These are certificates that have not been digitally signed by a CA, which is a method of confirming the identity of the certificate that is being served by the web server. They are signed by themselves, hence the name self-signed.	Test, dev or internal servers <b>only</b> .	1 - 13
CA-signed	A certificate that has had its identity digitally signed by a Certificate Authority (CA). This will allow browsers and clients to trust the certificate.	Production servers.	1 - 19

Digital Certificates that are issued by trusted 3rd party CAs (Certification Authority) provide verification that your Website does indeed represent your company, thereby verifying your company's identity. Many CAs simply verify the domain name and issue the certificate. Other CAs, such as [VeriSign](#), verify the existence of your business, the ownership of your domain name, and your authority to apply for the certificate, providing a higher standard of authentication.

A list of CA's can be found [here](#). Some of the most well known CAs are:

- [Verisign](#)
- [Thawte](#)
- [CAcert](#) (relatively new CA, providing free CA certificates)

We recommend using a CA-signed certificate.



If you're unable to install Portecle on the server or prefer the command line, see our [Command Line Installation](#) section below.

1. Download and install the [Portecle](#) app onto the server that runs Jira.

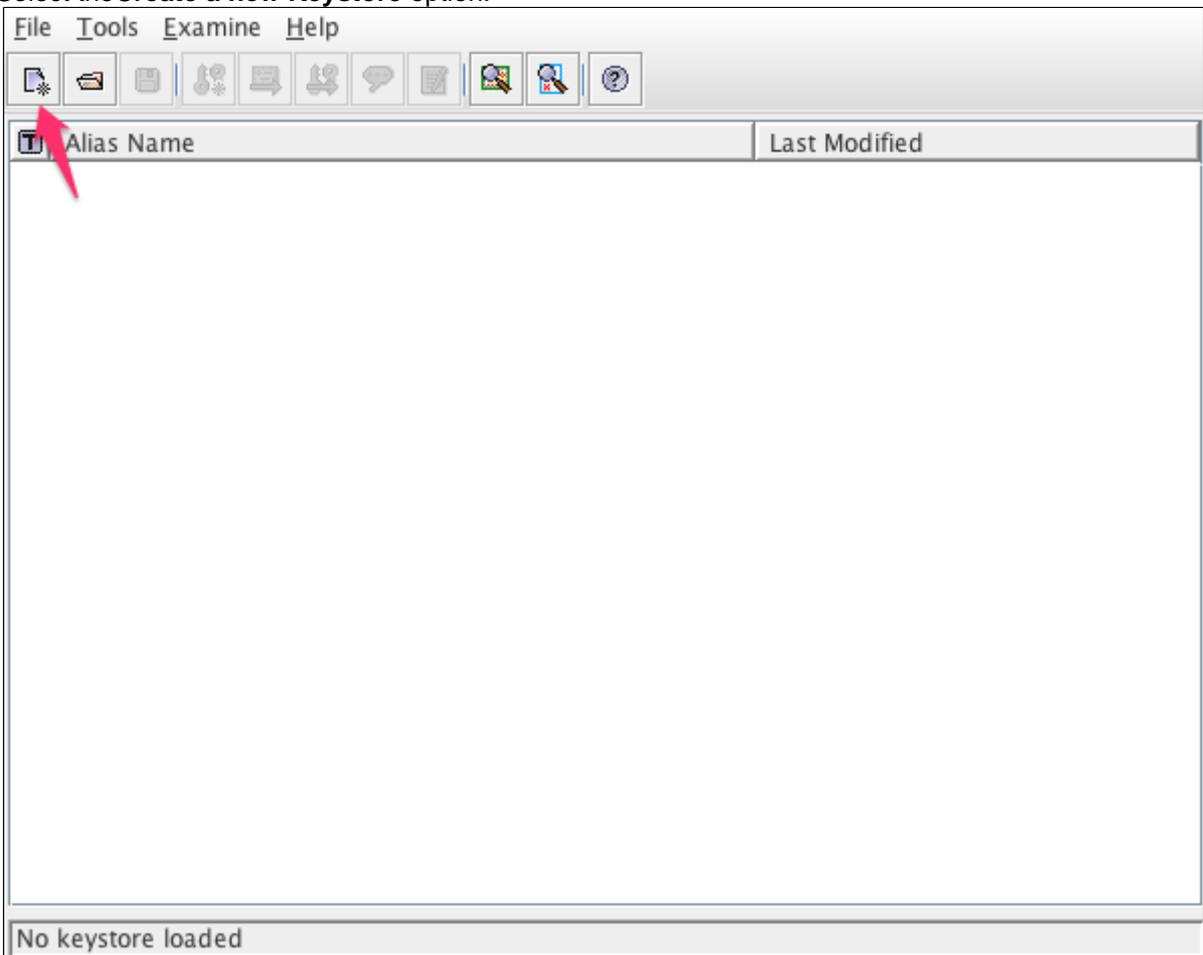
 This is a third-party application and is not supported by Atlassian.

2. Run the App as an Administrator, so it will have the appropriate permissions. Also, ensure the `<JAVA_HOME>` variable is pointing to the same version of Java that Jira uses. See [Setting JAVA\\_HOME](#) for further information on this.

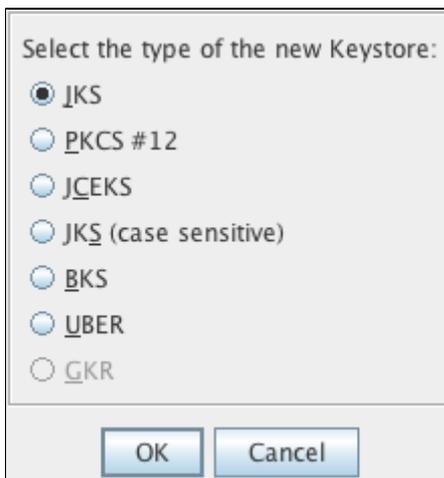
 If running on a Linux/UNIX server, X11 will need to be forwarded when connecting to the server (so you can use the GUI), as follows:

```
ssh -X user@server
```

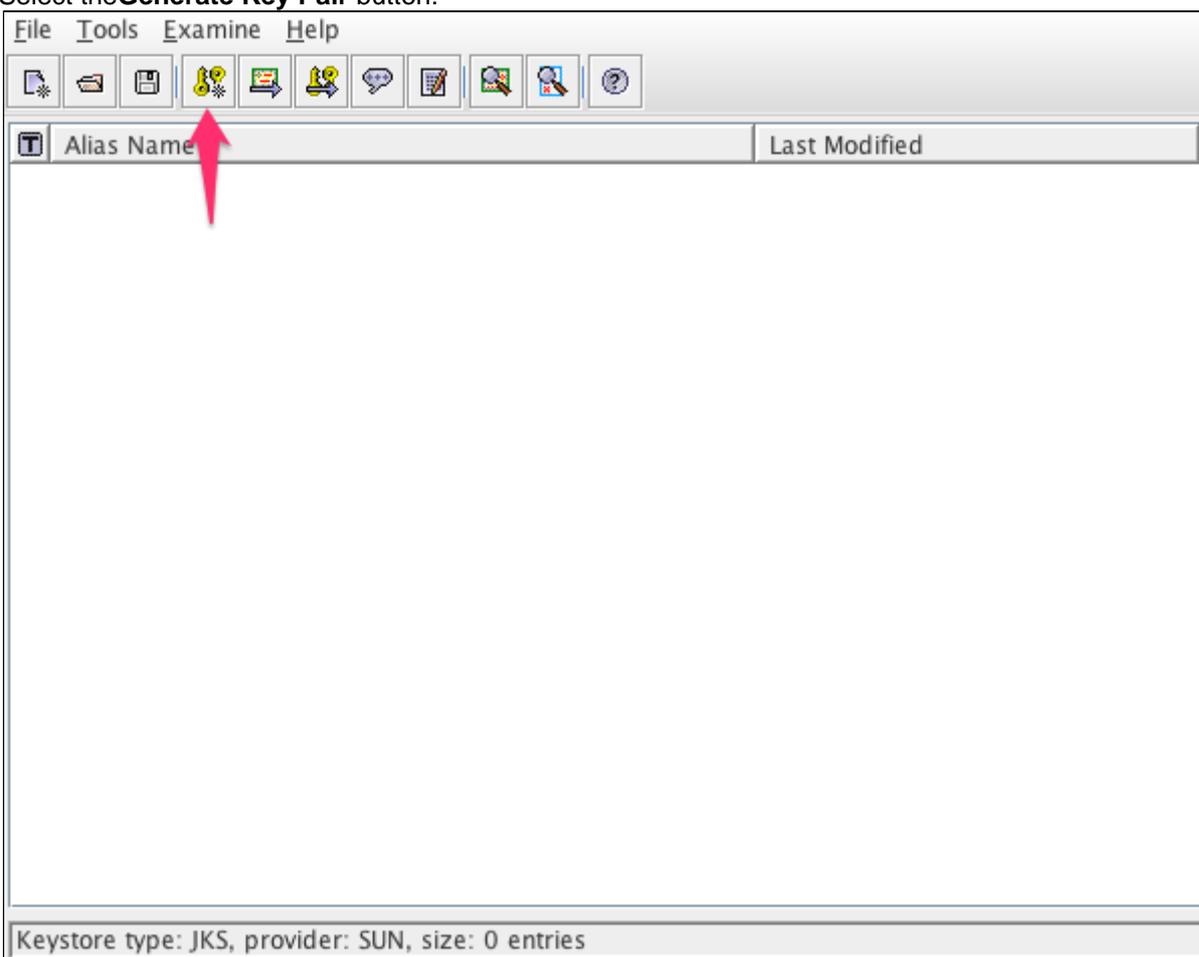
3. Select the **Create a new Keystore** option:



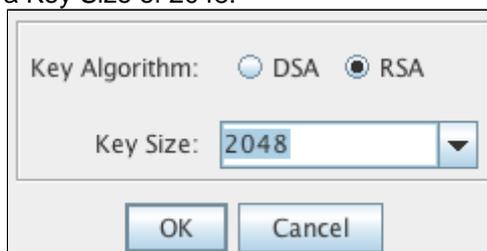
4. Select the type **JKS** and OK:



5. Select the **Generate Key Pair** button:



6. Select the RSA algorithm and a Key Size of 2048:



7. Make sure the **Signature Algorithm** is "SHA256withRSA" and refer to: [Security tools report the default SSL Ciphers are too weak.](#)

8. Edit the certificate details, as per the below example and select **OK**:

Signature Algorithm: SHA256withRSA

Validity (days): 365

Common Name (CN): jira.atlassian.com

Organisation Unit (OU): IT

Organisation Name (O): ATlassian Pty Ltd

Locality Name (L): Sydney

State Name (ST): NSW

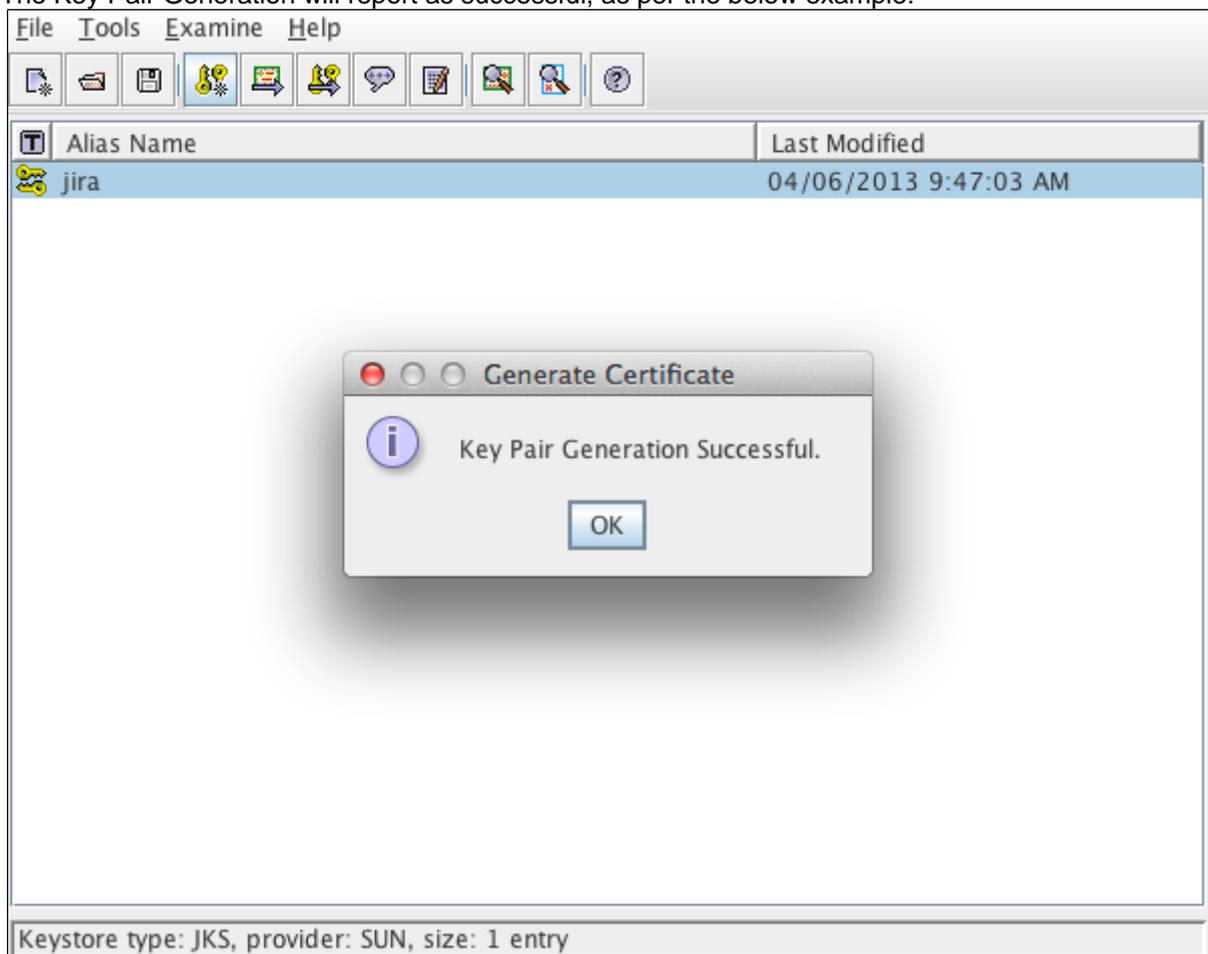
Country (C): US

Email (E): admin@example.com

OK Cancel

**!** The **Common Name** MUST match the server's URL, otherwise errors will be displayed in the browser.

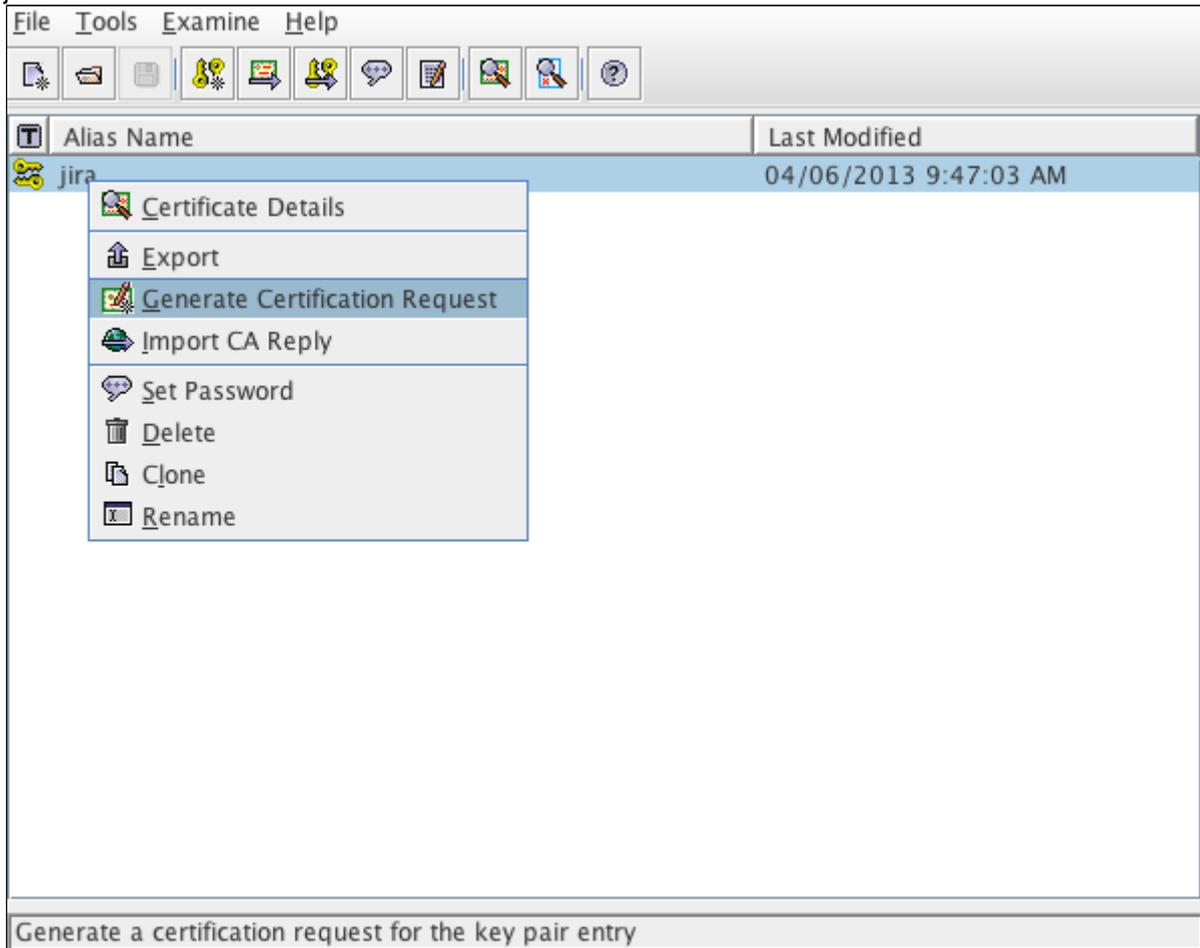
9. Choose an alias for the certificate - for example jira.  
 10. Enter a password for the KeyStore (the default password used is typically changeit).  
 11. The Key Pair Generation will report as successful, as per the below example:



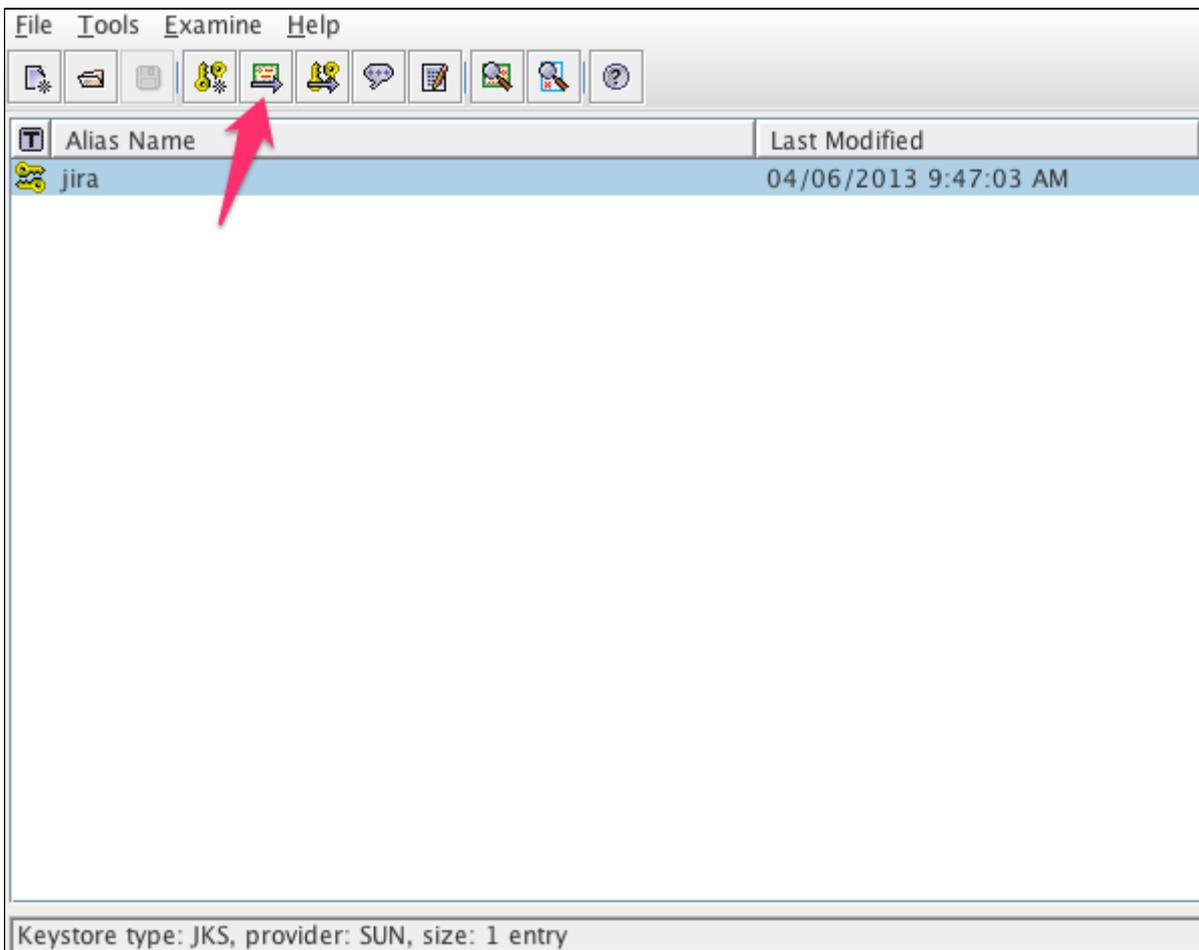
12. Save the KeyStore in <Jira\_HOME>/jira.jks, ensuring the use the same password in step 11. This can be done by **File > Save Keystore**.

**!** If using a self-signed certificate, proceed to [Configuring your web server using the Jira configuration tool](#). Otherwise, continue on.

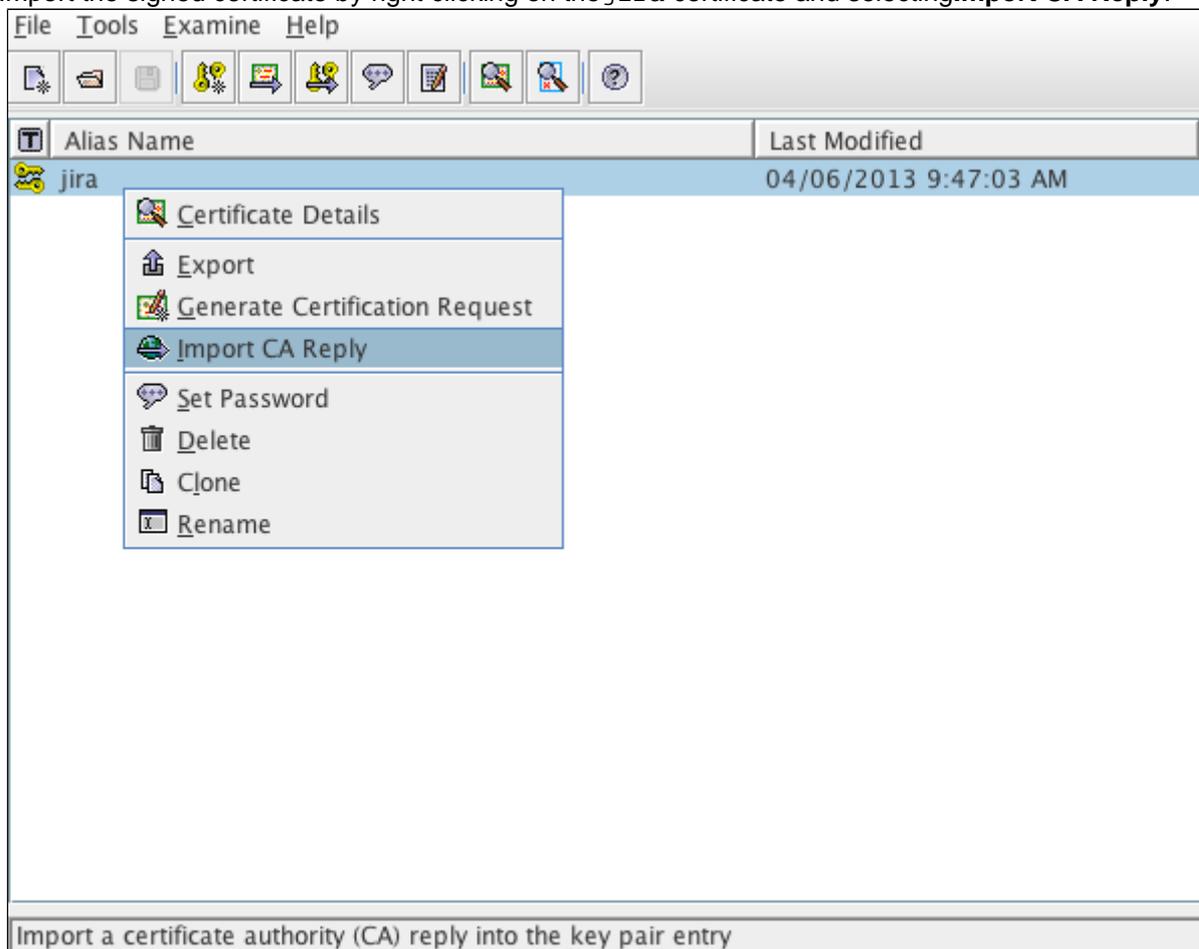
13. We need to generate a Certificate Signing Request for the CA to sign and confirm the identity of the certificate. To do so, right click on the certificate and choose **Generate CSR**. Save it in <Jira\_HOME>/jira.csr.



14. Submit the CSR to a Certificate Authority for signing. They will provide a signed certificate (CA reply) and a set of root/intermediate CA certificates.
15. Import the root and/or intermediate CA certificates with **Import Trusted Certificate**, repeating this step for each certificate.



16. Import the signed certificate by right clicking on the `jira` certificate and selecting **Import CA Reply**:



17. Select the certificate provided by the CA, which should be `jira.crt`. This will respond with CA Reply Import successful.
18. Verify this by checking **Tools > Keystore Report**. It should display the certificate as a child of the root certificates.
19. Save the KeyStore and proceed to the next section.

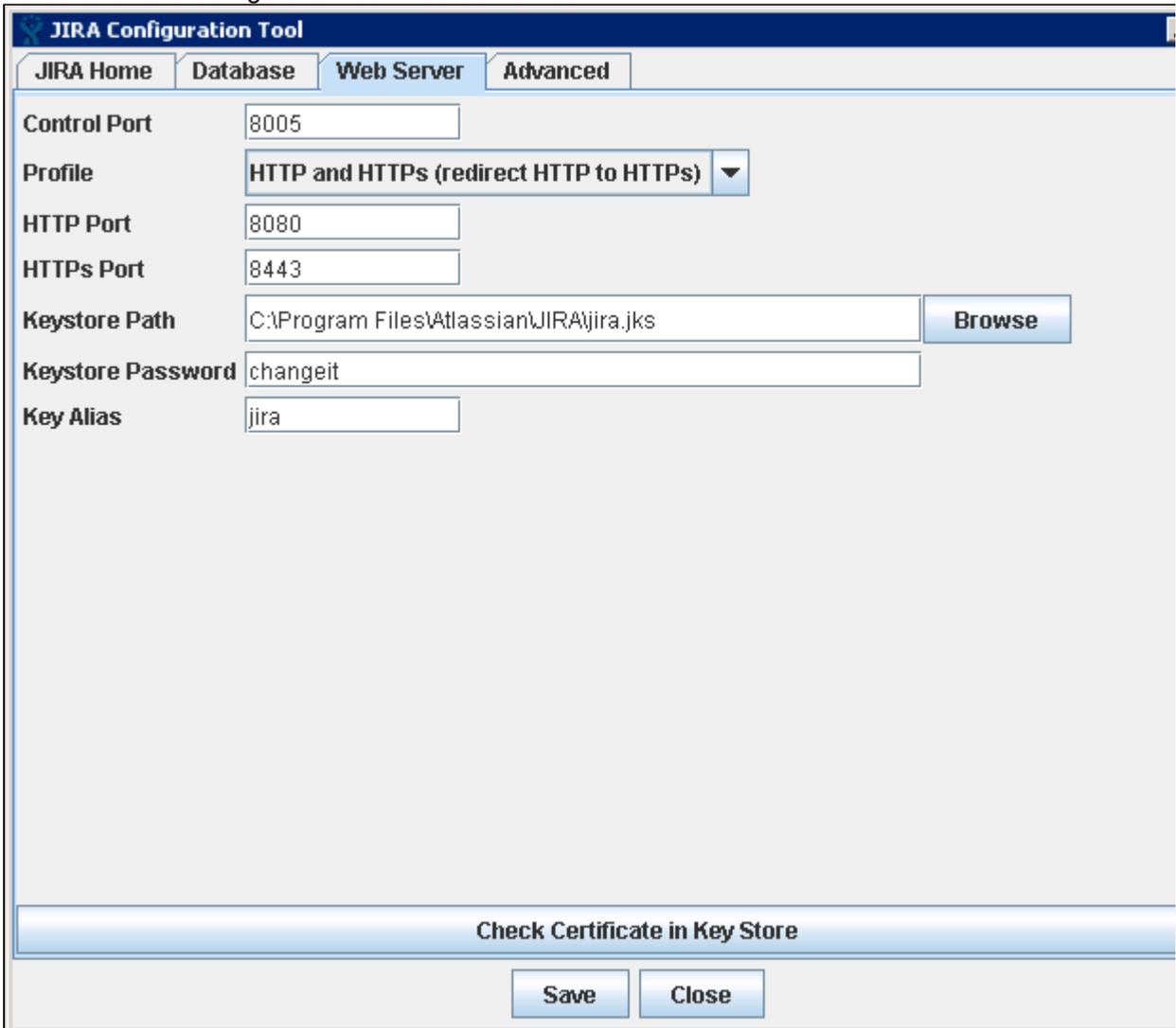
### Configuring your web server using the Jira configuration tool

In this section, you will finish setting up SSL encryption for Jira, by configuring your web server using the Jira configuration tool. For more information on the Jira configuration tool, see [Using the Jira configuration tool](#).

1. Run the Jira configuration tool, as follows:
  - **Windows:** Open a command prompt and run `config.bat` in the `bin` sub-directory of the [Jira installation directory](#).
  - **Linux/Unix:** Open a console and execute `config.sh` in the `bin` sub-directory of the [Jira installation directory](#).

**i** This command might fail with the error as described in [Unable to Start Jira applications Config Tool due to No X11 DISPLAY variable was set error](#). If it happens, refer to this article for the workaround.

2. Click the **Web Server** tab.  
 Screenshot: Jira configuration tool 'Web Server' tab



3. Fill out the fields as follows:

Field	Value

<b>Control Port</b>	Leave as default. You can change the port number if you wish. See <a href="#">Changing Jira's TCP ports</a> .
<b>Profile</b>	<p>A profile is a preset web server configuration. You can pick from the four following values:</p> <ul style="list-style-type: none"> <li>• Disabled</li> <li>• HTTP only</li> <li>• HTTP &amp; HTTPS (redirect HTTP to HTTPS)</li> <li>• HTTPS only</li> </ul> <p>To run Jira over HTTPS, you must pick either 'HTTP &amp; HTTPS' or 'HTTPS'. Pick 'HTTP &amp; HTTPS' if you want to run Jira over HTTPS but you have users that access Jira via HTTP. If you pick 'HTTP &amp; HTTPS', users who try to access Jira via HTTP will be redirected to the HTTPS address.</p>
<b>HTTP port</b>	Leave as default (8080). You can change the port number if you wish. See <a href="#">Changing Jira's TCP ports</a> . This will be disabled if you set the <b>Profile</b> to 'HTTPS only'.
<b>HTTPS port</b>	Leave as default (8443). You can change the port number if you wish. See <a href="#">Changing Jira's TCP ports</a> .
<b>Keystore path</b>	Specify the location of the keystore of your certificate. This will have been chosen when the keystore was saved in step 13 and should be <code>&lt;Jira_HOME&gt;/jira.jks</code> .
<b>Keystore password</b>	Specify the password for your keystore. If you generated a self-signed certificate, this is the password you specified for the key and keystore when generating the certificate in step 13.
<b>Keystore alias</b>	Each entry in the keystore is identified by an alias. We recommend using <code>jira</code> for this certificate as in step 10.

4. Select the **Check Certificate in Key Store** button to validate the following:
  - Test whether the certificate can be found in the key store.
  - Test whether keystore password works.
  - Test whether key can be found using key alias.
5. Save your changes.



When adding a new connection, the config tool doesn't include properties that allow special characters, so you'll need to add them manually to the `server.xml` file. For more info on how to do this, see [this kb article](#).

## Advanced configuration

### Running more than one instance on the same host

When running more than one instance on the same host, it is important to specify the *address* attribute in the `<Jira_INSTALLATION>/conf/server.xml` file because by default the connector will listen on **all available network interfaces**, so specifying the address will prevent conflicts with connectors running on the same default port. See the Tomcat Connector documentation for more about setting the address attribute in [The HTTP Connector](#) Apache Tomcat docs.

### Command Line Installation

#### Step 1. Create the KeyStore

1. Generate the Java KeyStore:

```
<JAVA_HOME>/keytool -genkey -alias jira -keyalg RSA -keystore <Jira_HOME>/jira.jks
```



 Instead of first and last name, enter the server URL, excluding "https://" (e.g.: [jira.atlassian.com](https://jira.atlassian.com)).

2. Enter a password.
3. Create the CSR for signing, using the password from step 2:

```
<JAVA_HOME>/keytool -certreq -alias jira -file /output/directory/csr.txt -keystore <Jira_HOME>/jira.jks
```

4. Submit the CSR to the CA for signing. They will provide a signed certificate and a root and/or intermediate CA.

 If the certificate will not be signed, skip to step 6.

5. Import the root and/or intermediate CA:

```
<JAVA_HOME>/keytool -import -alias rootCA -keystore <Jira_HOME>/jira.jks -trustcacerts -file root.crt
```

6. Import the signed certificate (the CA provides this):

```
<JAVA_HOME>/keytool -import -alias jira -keystore <Jira_HOME>/jira.jks -file jira.crt
```

7. Verify the certificate exists within the KeyStore:

```
<JAVA_HOME>/keytool -list -alias jira -keystore <Jira_HOME>/jira.jks
```

This must be a `PrivateKeyEntry`, if it is not the certificate setup has not successfully completed. For example:

```
jira, Jan 1, 1970, PrivateKeyEntry,
Certificate fingerprint (MD5): 73:68:CF:90:A8:1D:90:5B:CE:2A:2F:29:21:C6:B8:25
```

## Step 2. Update Tomcat with the KeyStore

1. Create a backup of `<Jira_INSTALL>/conf/server.xml` before editing it.
2. Edit the HTTPS connector so that it has the parameters that point to the KeyStore:

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
    maxHttpHeaderSize="8192" SSLEnabled="true"
    maxThreads="150" minSpareThreads="25"
    enableLookups="false" disableUploadTimeout="true"
    acceptCount="100" scheme="https" secure="true"
    sslEnabledProtocols="TLSv1.2,TLSv1.3"
    clientAuth="false" useBodyEncodingForURI="true"
    keyAlias="jira" keystoreFile="<Jira_HOME>/jira.jks" keystorePass="changeit"
    keystoreType="JKS"/>
```

 Make sure to put the appropriate path in place of `<Jira_HOME>` and change the port as needed.

If the organization doesn't support the latest TLS version, you can fallback to an earlier version. Change:

```
sslEnabledProtocols="TLSv1.2,TLSv1.3"
```

To:

```
sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2,TLSv1.3"
```

### 3. Edit the HTTP connector so that it redirects to the HTTPS connector:

```
<Connector acceptCount="100" connectionTimeout="20000" disableUploadTimeout="true" enableLookups="false" maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25" port="8080" protocol="HTTP/1.1" redirectPort="<PORT_FROM_STEP_1>" useBodyEncodingForURI="true"/>
```

**i** Ensure that the `<PORT_FROM_STEP_1>` is changed to the appropriate value. In this example it would be 8443.

### 4. Save the changes to `server.xml`.

### 5. If redirection to HTTPS will be used (this is recommended), edit the `<Jira_INSTALL>/WEB-INF/web.xml` file and add the following section at the end of the file, before the closing `</web-app>`. In this example, all URLs except attachments are redirected from HTTP to HTTPS.

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>all-except-attachments</web-resource-name>
    <url-pattern>*.jsp</url-pattern>
    <url-pattern>*.jspx</url-pattern>
    <url-pattern>/browse/*</url-pattern>
    <url-pattern>/issues/*</url-pattern>
  </web-resource-collection>
  <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
```

### 6. Restart Jira after you have saved your changes.

You can also redirect users from HTTP URLs to HTTPS URLs by choosing the 'HTTP & HTTPS' profile in the Jira configuration tool. This will redirect all HTTP URLs to HTTPS URLs.

If you want to only redirect certain pages to HTTPS, you need to do this manually. To do this, select the 'HTTPS only' profile in the Jira configuration tool and save the configuration, and then create an `htaccess` file on your web server that will manually redirect the HTTP URLs to the corresponding HTTPS URLs.

## Troubleshooting

Here are some troubleshooting tips if you are using a self-signed key created by Portecle, as described above.

When you enter `https://localhost:<port number>` in your browser, if you get a message such as "Cannot establish a connection to the server at localhost:8443", look for error messages in your `logs/catalina.out` log file. Here are some possible errors with explanations.

- **SSL + Apache + IE problems:** Some people have reported errors when uploading attachments over SSL using IE. This is due to an IE bug, and can be fixed in Apache by setting:

```
BrowserMatch ".MSIE." \
nokeepalive ssl-unclean-shutdown \
downgrade-1.0 force-response-1.0
```

[Google](#) has plenty more on this.

- **Can't find the keystore:**

```
java.io.FileNotFoundException: /home/user/.keystore (No such file or directory)
```

This indicates that Tomcat cannot find the keystore. The keytool utility creates the keystore as a file called `.keystore` in the current user's home directory. For Unix/Linux the home directory is likely to be `/home/<username>`. For Windows it is likely to be `C:\Documents And Settings\<UserName>`.

Make sure you are running Jira as the same user who created the keystore. If this is not the case, or if you are running Jira on Windows as a service, you will need to specify where the keystore file is in `conf/server.xml`. Add the following attribute to the connector tag you uncommented:

```
keystoreFile="<location of keystore file>"
```

This can also happen ("Cannot find /root/.keystore") if you add a `keystoreFile` attribute to the `http` connector in `server.xml` instead of the `https` connector.

- **Certificate reply and certificate in keystore are identical:**

```
keytool error: java.lang.Exception: Certificate reply and certificate in keystore are identical
```

This error will happen if you have identical names or fingerprints, which is the result of attempting to recreate the cert in your existing keystore. If you need to recreate or update the Cert, you may remove the existing keystore and creating a fresh, new keystore. In this case, creating a new keystore and adding the related certs will fix the issue. The default path for it in this documentation is `$JAVA_HOME/jre/lib/security/cacerts`

- **Incorrect password:**

```
java.io.IOException: Keystore was tampered with, or password was incorrect
```

You used a different password than "changeit". You must either use "changeit" for both the keystore password and for the key password for Tomcat, or if you want to use a different password, you must specify it using the `keystorePass` attribute of the Connector tag, as described above.

- **Passwords don't match:**

```
java.io.IOException: Cannot recover key
```

You specified a different value for the keystore password and the key password for Tomcat. Both passwords must be the same.

- **Wrong certificate:**

```
javax.net.ssl.SSLException: No available certificate corresponds to the SSL cipher suites which are enabled.
```

If the Keystore has more than one certificate, Tomcat will use the first returned unless otherwise specified in the SSL Connector in `conf/server.xml`.

Add the `keyAlias` attribute to the Connector tag you uncommented, with the relevant alias, for example:

```
<Connector port="8443" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true" useBodyEncodingForURI="true"
acceptCount="100" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="/opt/local/.keystore"
keystorePass="removed"
keyAlias="tomcat"/>
```

- **Using Apache Portable Runtime:**

APR uses a different SSL engine, and you will see an exception like this in your logs

```
SEVERE: Failed to initialize connector [Connector[HTTP/1.1-8443]]
LifecycleException: Protocol handler initialization failed: java.lang.Exception: No
Certificate file specified or invalid file format
```

The reason for this is that the APR Connector uses OpenSSL and cannot use the keystore in the same way. You can rectify this in one of two ways:

- Use the `Http11NioProtocol` to handle SSL connections Edit the `server.xml` so that the `SSL Connector` tag you just uncommented specifies the `Http11NioProtocol` instead of the APR protocol

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
  maxHttpHeaderSize="8192" SSLEnabled="true" keystoreFile="{user.home}/.keystore"
  maxThreads="150" enableLookups="false" disableUploadTimeout="true"
  acceptCount="100" scheme="https" secure="true"
  clientAuth="false" sslProtocol="TLS" useBodyEncodingForURI="true" />
```

- Configure the Connector to use the APR protocol This is only possible if you have PEM encoded certificates and private keys. If you have used OpenSSL to generate your key, then you will have these PEM encoded files - in all other cases contact your certificate provider for assistance.

```
<Connector
  port="8443" maxThreads="200"
  scheme="https" secure="true" SSLEnabled="true"
  SSLCertificateFile="{user.home}/certificate.pem"
  SSLCertificateKeyFile="{user.home}/key.pem"
  clientAuth="optional" SSLProtocol="TLSv1.2,TLSv1.3" />
```

- **Enabling Client Authentication:** To enable client authentication in Tomcat, ensure that the value of the `clientAuth` attribute in your `Connector` element of your Tomcat's `server.xml` file is `true`.

```
<Connector
  ...
  clientAuth="true"
  ... />
```

For more information about `Connector` element parameters, please refer to the [SSL Configuration HOW-TO Tomcat 8](#) documentation.

# Configuring security in the external environment

If your Jira instance contains sensitive information, you may want to configure security in the environment in which your Jira instance is running. Some of the main areas to consider are:

- Database:
  - If you are using an [external database](#), as recommended for production systems (i.e. you are not using Jira's internal/bundled H2 database), you should restrict access to the database that your Jira instance uses.
  - If you are using Jira's internal/bundled H2 database, you should restrict access to the directory in which you [installed](#) Jira. (Note that the user which your Jira instance is running as will require full access to this directory.)
- SSL if you are running your Jira instance over the Internet, you may want to consider using SSL.
- File system you should restrict access to the following directories (but note that the user which your Jira instance is running as will require full access to these directories):
  - [Index directory](#)
  - [Attachments directory](#)

## Other security resources

- [Jira application cookies](#)
- [User management](#)
- [Securing Jira applications with Apache HTTP Server](#)
- [Configuring permissions](#)

# Data collection policy

## Why does Jira collect usage data?

We're proud that Jira is one of the most advanced and configurable issue trackers on the planet and we will continue to deliver innovative new features as quickly as we can. In order to prioritize the features we deliver, we need to understand how our customers use Jira, what's important, what's not, and what doesn't work well. The collection of usage data allows us to measure the user experience across many thousands of users and deliver features that matter.

## What data is collected?

The type of data we collect is covered in our [Privacy Policy](#). Please read it, as we've tried to avoid legal jargon and make it as straightforward as possible.

To view a sample of data that might be collected from your specific installation:

1. Log in as a user with the **Jira Administrators** [global permission](#).
2. Choose **Administration**  **> System > Advanced > Analytics**.
3. Select the **Sample Data** link.

## How is data collected from Server installations?

Analytics are collected using the Atlassian Analytics app. The app collects analytics events in a log file which is located in the Jira home directory under the analytics-logs sub directory. The logs are periodically uploaded using an encrypted session and then deleted. If the Jira installation is unable to connect to the Internet, no logs are ever uploaded.

## Enabling/disabling data collection in Jira Server

You can switch off analytics collection at any time:

1. Log in as a user with the **Jira Administrators** [global permission](#).
2. Choose **Administration**  **> System > Advanced > Analytics**.
3. Select **Disabled**, and **Save** your change.

# Jira Admin Helper

The Jira Admin Helper is a **free, bundled app** that answers questions like:

- Why isn't my field showing up on view/edit/create screens?
- Why can/can't a user see a certain issue?
- Why did/didn't a user get a certain email notification?

**i** The Jira Admin Helper app is visible only to Jira Administrators. When you are viewing an issue, it is available from the **Admin** menu.

**On this page:**

- [Field Helper](#)
- [Permission Helper](#)
- [Notification Helper](#)

## Field Helper

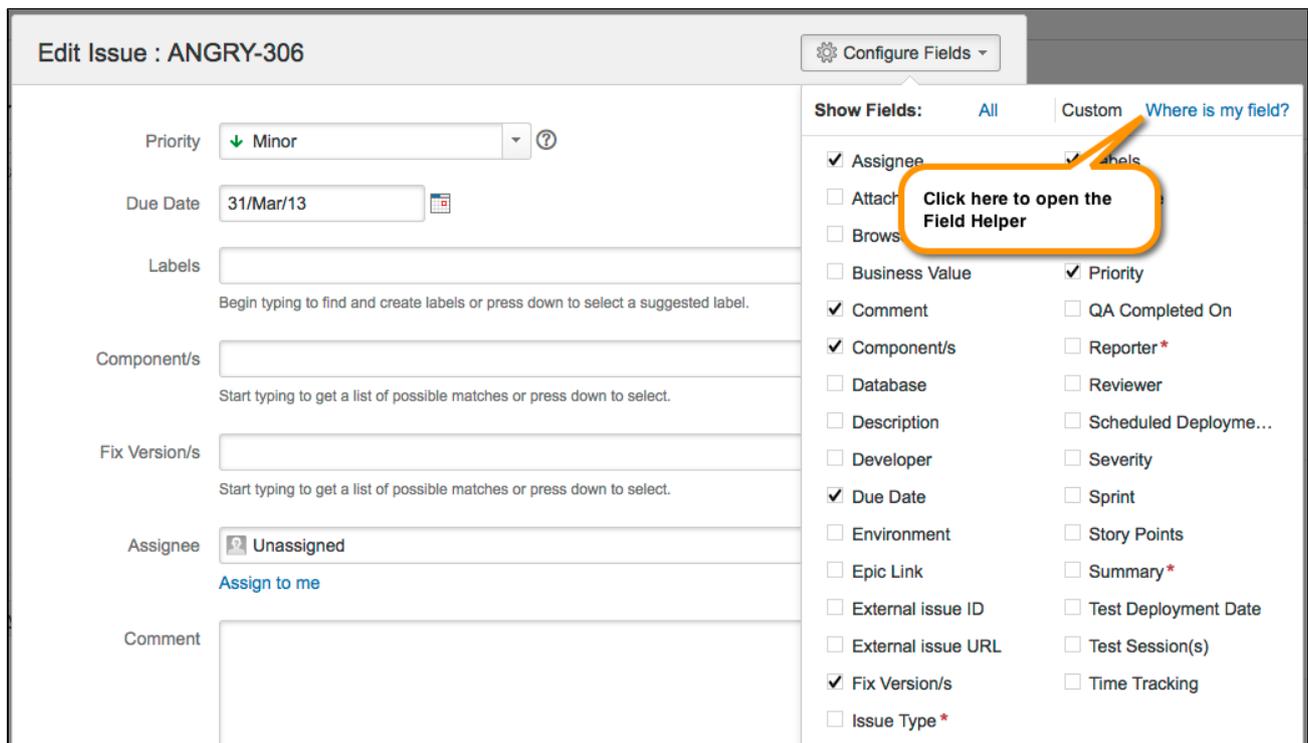
If you're logged in as a Jira administrator, you can use the Field Helper displayed as a *Where is my field?* link to help you determine why a field is not appearing on a specific screen. The Field Helper works with custom fields as well as Jira system fields.

The *Where is my field?* link is available on:

- Create Issue in *Configure Fields* pop up
- Edit issue - in *Configure Fields* pop up
- View Issue - in *More Actions* menu
- Issue Navigator in cog menu

Simply click on the link and then enter the field name in the search box!

Here's an example:



After you enter the name of the missing field, the Field Helper returns a form that explains why this field is not appearing:

### Where is my field?

Begin typing to find your field

Project: **Angry Nerds**  
 Issue type: **Bug**  
 Screen: **Edit Issue**  
 Field: **Need Docs?**  
 Status: ✘

The 'Need Docs?' field is not present on the form you are viewing

Status	Summary	Details
<span style="color: green;">✔</span>	Field configuration	The field 'Need Docs?' is enabled by the Field Configuration 'Angry Nerds Field Configuration' associated with this issue.
<span style="color: green;">✔</span>	Field Screen	The field 'Need Docs?' is present on the screen 'Default Screen' configured for this issue
<span style="color: red;">✘</span>	Project and issue type scope	The field 'Need Docs?' is not configured in scope of the project 'Angry Nerds' and issue type 'Bug' To solve this issue, go to <a href="#">'Need Docs?' configuration page</a> and add it to the scope

[Close](#)

You can then use this information to fix your screen by adding this field to your project and issue type.

## Permission Helper

The Jira Admin Helper can help you diagnose why a user can or cannot see a certain issue.

1. Choose **Administration** () > **System**.
2. Then choose **Admin Helper** > **Permission Helper**.
3. Enter the username of the user (leave blank for anonymous users), an issue key (for example, an issue that the user can/cannot see) and the permission to check.
4. Click **Submit**.

**Permission Helper**

Discover why a user does or does not have certain permissions...

User

Begin typing to find a user, leave blank for Anonymous user

Permission

Begin typing to find a permission or press down to see all

## Notification Helper

The Notification Helper can you help figure out why a user didn't get an email notification when a comment was added. It's available from the view issue page, the issue navigator, and from Jira Administration.

1. Choose **Administration** (⚙️) > **System**.
2. Then choose **Admin Helper** > **Notification Helper**.
3. Enter the username of the user (leave blank for anonymous users) and select the Notification Event from the drop-down list.
4. Click **Submit**.

**Notification Helper**

Find out why users receive, or do not receive notifications for this issue

User

Begin typing to find a user

Notification Event

Begin typing to find a notification event or press down to see all

# Raising support requests as an administrator

If you have a problem with your Jira instance, there are a number of resources available to help you resolve it. We recommend that you try searching our knowledge base and our customer forums for an answer first. This is often the fastest way to get a problem resolved.

- [Jira knowledge base](#)
- [Atlassian Answers](#)

If you can't find what you need, the next step is to raise a support request, as described below.

## On this page:

- [Before you begin](#)
- [Raising a support request in Jira](#)
- [Creating a support zip](#)
- [Providing logs when you cannot log in to Jira](#)

## Before you begin

The functionality described on this page is enabled by the [The Troubleshooting and Support Tools plugin](#), which is bundled with Jira. This functionality is only available to users with the **Jira System Administrators** global permission.

If you do not have this permission, you can still raise support requests on our [support site](#). You'll need to provide as much information as possible, including:

- Any error messages that are appearing on the console or in the logs (see the [logging](#) documentation).
- The operating system, database and version of Jira you are using.

## Raising a support request in Jira

[Raising a support request](#) in Jira gives you the option of including a range of system information with your request, rather than you having to manually describe it. This information helps our support team resolve your issue faster.

1. Log in as a user with the **Jira System Administrators** global permission.
2. Click **Administration** (⚙️) > **System** > **Troubleshooting and Support tools** > **Get Help** > **Contact Technical Support or Report a Bug**.
3. Fill out the **Create Support Request** form. Include as much information as possible to help our support team resolve your issue faster.

# Troubleshooting and support toc

Identify, diagnose and solve problems with your help, contact Atlassian Support using the support

[Instance health](#)   [Log analyzer](#)   [Get help](#)   [C](#)

[Get help](#) » Create support request

## Create support request

To **Atlassian Support Team**

Contact email\*

Summary\*

Description\*

Severity

Attach Support Zip (cu

Send

Cancel

4. Click **Create**.

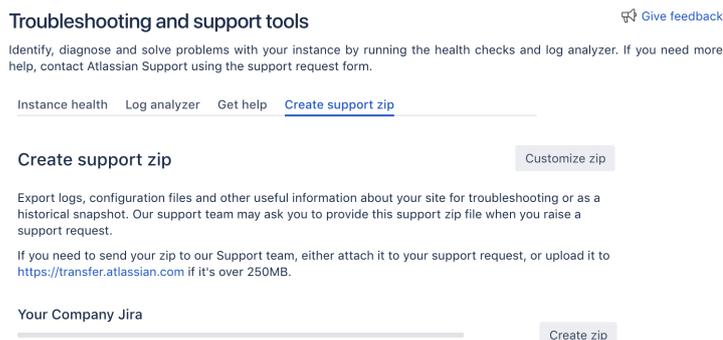
Once you've submitted your support request, we'll send you email updates about its progress.

## Creating a support zip

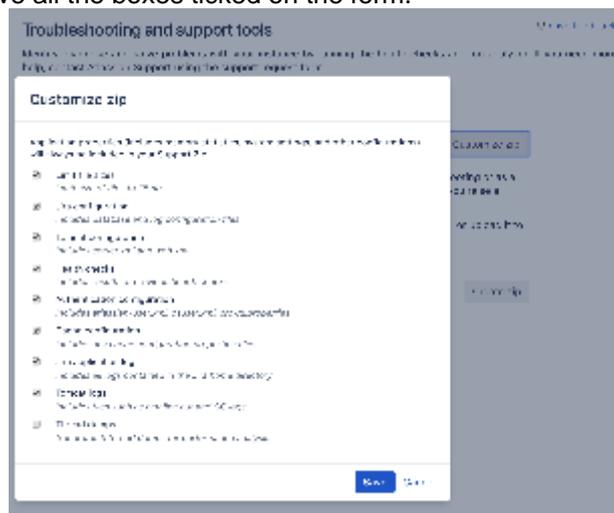
The Atlassian Support team may need you to provide a support zip to help them understand and troubleshoot your problem, after you have raised your initial request. The support zip contains logs from your instance and diagnostic and configuration information.

Note that the Support Tools Plugin sanitizes any usernames and passwords it finds in your configuration files, but does not sanitize usernames and content present in the log files.

1. Log in as a user with the **Jira System Administrators** global permission.
2. Click **Administration**  **> System > Atlassian Troubleshooting and Support tools > Create support zip**.



3. (Optional) If you need to customize your zip, click the **Customize zip** button. However, we recommend that you leave all the boxes ticked on the form.



4. Click **Create zip**.
5. Click **Download** to download the Support Zip to your local.

You can now provide this to the Atlassian Support team (usually by attaching the Support Zip to your support request).

## Providing logs when you cannot log in to Jira

If you are unable to log in to Jira, you can still provide helpful diagnostic information to the Atlassian Support team. Create compressed files (zip or tar.gz) of the following log files and attach them to your support request:

- **Latest Jira logs:** `$Jira_HOME/log/atlassian-jira.log`
- **Application server (Tomcat) log files:**
  - UNIX: `$Jira_INSTALL/logs/catalina.out`
  - Windows: `$Jira_INSTALL/logs/stdoutandstderr`

If you cannot locate these files, compress the contents of the following directories and attach them to your support request:

1. `$Jira_INSTALL/logs`
2. `$Jira_HOME/log`

# Start and Stop Jira applications

How you start and stop your Jira application depends on whether you are running Jira as a Service.

## Windows

If you installed Jira as a service, you can **Start Jira Server** and **Stop Jira Server** from the Windows Start menu.

You can't start or stop Jira manually using the `start-jira.bat` and `stop-jira.bat` file.

If you didn't install Jira as a service you'll need to start and stop Jira manually.

- To start Jira run `<installation-directory>\bin\start-jira.bat`
- To stop Jira run `<installation-directory>\bin\stop-jira.bat`

We recommend running Jira with a dedicated user account. To do this, use the `runas` command to execute `start-jira.bat`.

```
> runas /env /user:<DOMAIN>\<jira> start-jira.bat
```

Where `<DOMAIN>` is your Windows domain or computer name and `<jira>` is the name of your dedicated user.

Jira can be started with all non-system apps, or a selection of these apps, disabled. This helps with troubleshooting when these non-system apps are causing issues with your Jira instance, such as causing Jira to fail on start up, or when the app is malfunctioning and can't be removed through UPM. The parameters do not persist, that is, they are applied at start up once, and if you need to restart Jira, you need to apply them again.

The parameters should be specified at system start up when Jira is started using the `start-jira.bat` file, for example:

```
<installation-directory>/bin/start-jira.bat /disablealladdons /disableaddons=com.atlassian.test.plugin
```

If you don't use `start-jira.bat` for starting Jira, but still wish to use this feature, you can add the following JVM parameter to the invocation of the servlet container that's running Jira:

```
-Datlassian.plugins.startup.options="/disablealladdons /disableaddons=com.atlassian.test.plugin"
```

To disable **multiple apps**, use a colon separated list of apps. Regex/wildcards are not permitted, the full key of the app must be provided, for example:

```
-Datlassian.plugins.startup.options="/disablealladdons /disableaddons=com.atlassian.test.plugin:com.atlassian.another.test.plugin"
```

## Notes

- If the app key contains a space, this feature will not work, you need to [manually deal with that app](#).
- This feature does **not** work for Jira Data Center applications.
- This can be used to disable an app deemed critical to Jira starting, and if one of those is disabled, Jira will fail to start
- This can be used to disable Jira application OBR bundles, for example, to stop the Jira Software app:

```
<installation-directory>/bin/start-jira.bat /disableaddons=com.atlassian.jira.jira-software-application
```

## Linux

If you installed Jira as a service, use one of the following commands to **start** or **stop** Jira.

```
$ sudo /etc/init.d/jira start
$ sudo /etc/init.d/jira stop
```

You can't start or stop Jira manually using the `start-jira.sh` and `stop-jira.sh` files.

If you didn't install Jira as a service you'll need to start and stop Jira manually.

- To start Jira run `<installation-directory>\bin\start-jira.sh`
- To stop Jira run `<installation-directory>\bin\stop-jira.sh`

We recommend running Jira with a dedicated user account:

```
$ su -u <user>
$ ./start-jira.sh
```

Where `<user>` is the name of your dedicated user.

If you're using Ubuntu the command is a little different:

```
$ sudo su <user>
$ ./start-jira.sh
```

Jira can be started with all non-system apps, or a selection of these apps, disabled. This helps with troubleshooting when these non-system apps are causing issues with your Jira instance, such as causing Jira to fail on start up, or when the app is malfunctioning and can't be removed through UPM. The parameters do not persist, that is, they are applied at start up once, and if you need to restart Jira, you need to apply them again.

The parameters should be specified at system start up when Jira is started using the `start-jira.sh` script, for example:

```
./bin/start-jira.sh --disable-all-addons --disable-addons=com.atlassian.test.plugin
```

If you don't use `start-jira.sh` for starting Jira, but still wish to use this feature, you can add the following JVM parameter to the invocation of the servlet container that's running Jira:

```
-Datlassian.plugins.startup.options="--disable-all-addons --disable-addons=com.atlassian.test.plugin"
```

To disable **multiple apps**, use a colon separated list of apps. Regex/wildcards are not permitted, the full key of the app must be provided, for example:

```
-Datlassian.plugins.startup.options="--disable-all-addons --disable-addons=com.atlassian.test.plugin:com.atlassian.another.test.plugin"
```

## Notes

- `--disable-addons` takes a colon-separated list (chosen as a colon is the only prohibited character from an app key) of addons to be disabled. These **can** be system apps.
- This feature does **not** work for Jira Data Center applications.
- This can be used to disable an app deemed critical to Jira starting, and if one of those is disabled, Jira will fail to start
- This can be used to disable Jira application OBR bundles, for example, to stop the Jira Software app:

```
./bin/start-jira.sh --disable-addons=com.atlassian.jira.jira-software-application
```

# Managing LexoRank

LexoRank is ranking system that Jira Software uses which provides the ability to rank issues in Jira Software instances. The user interface can be found in **Administration (⚙️) > System > Lexorank management**. It provides a user interface for several of the key areas of LexoRank administration.

## Balancing

Balancing

LexoRank management consists of two parts: the LexoRank service that determines if a rebalance can be run; and the actual rebalance of the Rank field in the database. Note, if you are running clustered JIRA, a balance can only run on one node of the cluster at a time. Also, the service status will only show you activity on this node.

Database status at Tue Jun 20 2017 13:40:25 GMT+1000 (AEST) Refresh Balance all fields

Field name	Field id	Num ranked issues	Percent complete	Rank value distribution	Rank status ⓘ
Rank	10005	18279	100%	18279,0,0	Status: <span style="color: green;">OK</span> Length: 18 / 254 Next rebalance: Scheduled Issue with highest rank length: JSDS-693 <a href="#">Balance field</a>

Service status at Tue Jun 20 2017 13:40:25 GMT+1000 (AEST) Refresh

Balancing disabled	false
Balancing suspended	false
Balance handler running	false
Balancing in progress on a node of the cluster	false

As per the screenshot above, the UI provides the user with details on the distribution of issues in each bucket, and whether or not a balance is in progress or if it's disabled. Running a balance is similar to running a re-index of Jira, there isn't any guarantee it'll fix a problem unless the problem is specific to the ranking data.

The **Rank Status** column provides details about the status of the ranked issues. Each issue has a unique rank relative to the issues around it. This rank is stored as a string. As the amount of issues grows and you perform more ranking operations, the length of these strings increase. At some point, the length reaches a threshold, and **arebalance** is scheduled within the next 12 hours.

A rebalance will evenly distribute the ranked issues, and significantly reduce the rank length. During a rebalance, ranking operations can continue as normal.

Should the length reach a second threshold within 12 hours, an **immediate rebalance** is started.

Should the length reach a third threshold before rebalance is finished, all rank operations are **disabled** until the rebalance completes.

The **Rank Status** field has these properties:

Property	Possible values
Status	<ul style="list-style-type: none"> <li>OK - the rank length is in a healthy state</li> <li>Warning - a rebalance has been scheduled</li> <li>Critical - an immediate rebalance has started, you are approaching a state where rank operations will be disabled</li> </ul>
Length	<current length> / <maximum length>  Maximum length indicates when rank operations will be disabled
Next rebalance	<ul style="list-style-type: none"> <li>Scheduled - once the next threshold is reached, a rebalance will be scheduled</li> <li>Immediate - once the next threshold is reached, a rebalance will start immediately</li> </ul>

In addition to the above, the field will tell you which project has the issues with the longest rank. This can be useful to diagnose the cause of a rapid increase in rank length.

If you're encountering issues and don't know why, or are unsure whether or not to balance, review the integrity checks first and contact support if need be as per the below details.

A breakdown of the service status is below:

Service	Notes
Balancing disabled	<p>If this is <code>true</code> then Jira Software has disabled balancing internally:</p> <ul style="list-style-type: none"> <li>• A foreground re-index may be running. It is expected that this will disable balancing.</li> <li>• Jira Software may have just been installed / upgraded and requires a reindex.</li> <li>• Check the logs for any exceptions and see if there are existing KB articles for these errors.</li> <li>• See if there is anything failing in the Integrity checks.</li> </ul>
Balancing suspended	<ul style="list-style-type: none"> <li>• This will be false unless balancing has been explicitly suspended by Support or an administrator.</li> </ul>
Balance handler running	<ul style="list-style-type: none"> <li>• This will be <code>false</code> unless a balance is currently in progress. To verify the progress of the balance, hit the refresh button.</li> <li>• On a Jira Data Center cluster this will be true on the node that is running the balance, and false on the other nodes.</li> </ul>
Balancing in progress on a node of the cluster	<ul style="list-style-type: none"> <li>• It indicates that the balance cluster lock has been taken.</li> <li>• This will be <code>true</code> when a balance is running.</li> </ul>

## Integrity checks

Integrity checks

If you spot a ranking problem in the Balancing section above, run the integrity checks. The integrity checks generate a report for each rank field in the system. Failing integrity checks indicate ranking data may be corrupted, please contact support and provide details of failed checks.

Report for rank field Rank, FieldId=10005 at Tue Jun 20 2017 13:49:05 GMT+1000 (AEST)		
Marker rows present in table for rank field	<span style="color: green;">PASSED</span>	Checks if the rank table has been properly initialized for the rank field. A minimum and maximum marker row are expected to be present in the table for the rank field.
Marker rows correctness check	<span style="color: green;">PASSED</span>	Checks whether the marker rows for a rank field have the expected rank value.
Marker rows in valid bucket check.	<span style="color: green;">PASSED</span>	Checks if the marker rows of a rank field are in valid bucket(s).
Rank out of bounds check	<span style="color: green;">PASSED</span>	Checks if there is a rank value for the rank field that is out of bounds.
Issue ranks different from marker ranks check	<span style="color: green;">PASSED</span>	Checks if there are any issue ranks that have the same rank as the maximum or minimum rank.
Issue rows in valid bucket check.	<span style="color: green;">PASSED</span>	Checks if the issue rows of a rank field are in a valid bucket. Issue rows should be in one of the buckets the marker rows are in.
Balance status check	<span style="color: green;">PASSED</span>	Checks if there is a correct balance entry is present or absent for the rank field.
Bucket field check	<span style="color: green;">PASSED</span>	Checks that the bucket field in the rank table matches the bucket digit stored in the rank field for all valid bucket values
Duplicate ranks check	<span style="color: green;">PASSED</span>	Checks if there are any duplicate rank values for a rank field.

This runs a series of tests against the LexoRank data and returns a true / false result based on the test. The tests are detailed below.

Check	How to Fix Failures
Marker rows present in table for rank field.	If this fails, the minimum or maximum marker rows are missing.
Marker rows correctness check.	If this fails, the minimum or maximum marker rows exist, however they have the incorrect rank.  This can be fixed by updating the rank on the row returned in the check to be the expected value.
Marker rows in valid bucket check.	When a balance is in progress, the marker rows are moved to another bucket to indicate where the new rank values should be. The only time they should be in different buckets is if a balance is in progress.  Valid states for the marker rows are the below. <ul style="list-style-type: none"> <li>• Minimum is the same as maximum.</li> <li>• Minimum is 0, and max is 1.</li> <li>• Minimum is 1, and max is 2.</li> <li>• Minimum is 0, and max is 2.</li> </ul> This test fails if the marker rows are not in those buckets, and is likely caused by exceptions thrown during the rank creation or balance operation. Please check the logs for those and verify them against known problems.
Rank out of bounds check.	Please refer to our <a href="#">How to Fix Rank Out Bound Error</a> KB article for the fix.
Duplicate ranks check.	This can be fixed as detailed in <a href="#">How To Fix Duplicate Rank Values For a Rank Field</a> .
Issue ranks different from marker ranks check.	The below SQL will identify records that have ranking values the same as the min/marker rows. Deleting these records will correct this problem (and also result in a loss of ranking data for those issues only). <pre>SELECT * FROM "AO_60DB71_LEXORANK" WHERE ("RANK" LIKE '% zzzzzz:' OR "RANK" LIKE '% 000000:') AND "TYPE" not in (0,2);</pre> <p><b>i</b> This may require changing depending upon the DBMS used.</p>

Issue rows in valid bucket check.	If a balance can't fix this, it would need detailed analysis from Atlassian Support.
Balance status check.	Attempt a balance, and check the logs to see if there are any exceptions. It's likely the balance is failing due to a exception in the logs, or one of the other checks above may be failing.

If you're unsure how to proceed, please feel free to contact support for further assistance.

# Jira cluster monitoring

If your Jira Data Center is clustered, you can easily use Jira tools to know how the nodes are doing.

The new Cluster monitoring page available to Jira system administrators gathers real-time data such as node ID, address, uptime (since last restart), load (system load average for the last minute), and memory. The information available on the page can also help you decide if a node you've just added to the cluster has been configured correctly.

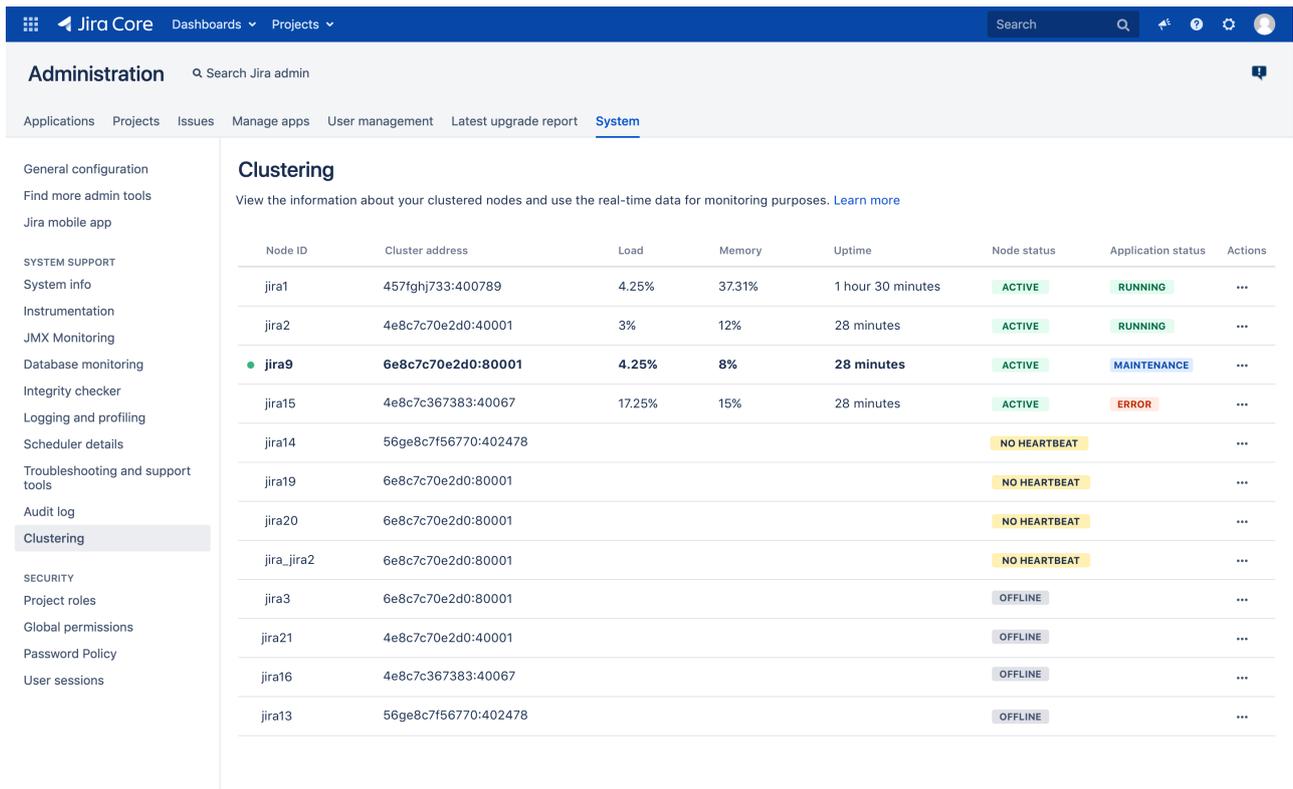
 This feature is only available in Jira Data Center.

## Skip directly to:

- [Monitor the status and health of your node](#)
- [View cluster information in the audit log](#)
- [View runtime and system info](#)
- [View the custom fields that take longest to index](#)

## View your clustered nodes

To see the information about your clustered nodes, go to **Jira Administration > System > Clustering**. For more on Jira clustering, see [Configuring a Jira cluster](#).



Node ID	Cluster address	Load	Memory	Uptime	Node status	Application status	Actions
jira1	457fghj733:400789	4.25%	37.31%	1 hour 30 minutes	ACTIVE	RUNNING	...
jira2	4e8c7c70e2d0:40001	3%	12%	28 minutes	ACTIVE	RUNNING	...
<b>jira9</b>	<b>6e8c7c70e2d0:80001</b>	<b>4.25%</b>	<b>8%</b>	<b>28 minutes</b>	ACTIVE	MAINTENANCE	...
jira15	4e8c7c367383:40067	17.25%	15%	28 minutes	ACTIVE	ERROR	...
jira14	56ge8c7f56770:402478				NO HEARTBEAT		...
jira19	6e8c7c70e2d0:80001				NO HEARTBEAT		...
jira20	6e8c7c70e2d0:80001				NO HEARTBEAT		...
jira_jira2	6e8c7c70e2d0:80001				NO HEARTBEAT		...
jira3	6e8c7c70e2d0:80001				OFFLINE		...
jira21	4e8c7c70e2d0:40001				OFFLINE		...
jira16	4e8c7c367383:40067				OFFLINE		...
jira13	56ge8c7f56770:402478				OFFLINE		...

The node marked with a green dot is the one you're currently on.

## Monitor the status and health of your node

- **ACTIVE**: the node is active and has heartbeat.

**i** If an **ACTIVE** node gets flagged as **OFFLINE** Jira automatically shuts down to prevent any possible damage to cluster. This is not an official and recommended way to shut down the Jira cluster. You can disable this feature by using the `jira.cluster.node.self.shutdown.if.offline.disabled=true` property.

- **NO HEARTBEAT**: the node is temporarily down, has been killed, or is active but failing. This state might be caused by automatic deployment. It might also happen that the server is starting and the No heartbeat status is temporary. Normally, a node is moved to this state after 5 minutes of reporting no heartbeat. You can check if a node is really down and decide if you want to restart it, or remove it right away through REST API.

**i** If a node is killed abruptly, it might still show as active for about 5 minutes before changing the status to No heartbeat.

By default, after two days if reporting the No heartbeat state the node is automatically moved offline. However, you can change the default by modifying the `jira.not.alive.active.nodes.retention.period.in.hours` system property. Alternatively, you can add a JVM flag on Jira startup. For example, if you want the node to go offline after 3 hours, enter the following flag:

```
Djira.not.alive.active.nodes.retention.period.in.hours=3
```

**i** The value for `jira.not.alive.active.nodes.retention.period.in.hours` should be greater than the Jira instance start-up time otherwise other nodes in the cluster could move the node to the **OFFLINE** state.

- **OFFLINE**: the node has been manually fixed or stopped, and moved offline. The node should not be used to run cluster jobs. Once moved offline, the node is automatically removed from cluster after two days.

**i** If you don't want the nodes to be automatically moved offline and removed from cluster, you can disable the feature by using the `jira.cluster.state.checker.job.disabled=true` property. Additionally, if you want to use your own scripts, you can use the API described [here](#).

### Application status

The responses are refreshed every minute on the server side. Refresh the page to get the latest data. If a node is offline or has no heartbeat, the Application status column does not contain any data.

**MAINTENANCE**: Jira on the node is being reindexed and cannot currently serve users.

**ERROR**: something went wrong on startup and Jira is not running on this node. The error might have been caused by multiple reasons such as: the database couldn't be reached, or a lock file is found. Check the log files for details.

**RUNNING**: the node is up and Jira is running on it.

**STARTING**: the node with Jira is starting up.

### Cluster information in the audit log

To help you manage your cluster, you can also find the information on nodes leaving or joining the cluster in your Audit log.

To collect these events, set the **Global configuration and administration** coverage area to **Full**:

1. Go to **Jira administration > System > Audit log**.
2. On the **Advanced audit log** admin page, click **Settings**.
3. Under **Coverage**, set the **Global configuration and administration** coverage area to **Full**, and then click **Save**.

To see the logged cluster-related events, go to **Jira administration > System > Audit log**, and then search for the "clustering" keyword.

We log the following events:

- NodeJoined - a new node has joined the cluster
- NodeReJoined - an existing node was restarted and re-joined the cluster
- NodeLeft - a node received the OFFLINE status
- NodeRemoved - a node was removed from the cluster
- NodeUpdated - all other cases when existing node was updated

## View runtime and system info

To drill down and see runtime and system information, click **Actions** for a specific node.

System information

← Back to clustering   ↻ Refresh   📄 Export to csv

Key	Value
atlassian.cache.ehcache	true
atlassian.cluster.scale	true
atlassian.dev.mode	true
atlassian.disable.caches	true
atlassian.disable.issue.collector	true
atlassian.disable.spring.cache.bean.metadata	false
atlassian.mail.fetch.disabled	true
atlassian.mail.send.disabled	true
atlassian.plugins.tenant.smart.patterns	/Users/hamilj@atlassian.com/works/jira/jira-components/jira-core/src/main/resources/tenant-smart-patterns.txt
atlassian.renderer.max.emojis	1000
atlassian.rest.filesize.exceeded.statuscode.legacy.enabled	true
atlassian.webservice.disable.minification	true
awt.toolkit	sun.awt.mac.osx.LWCToolkit
catalina.base	/Users/hamilj@atlassian.com/works/jira/target/tomcat/node3
catalina.home	/Users/hamilj@atlassian.com/works/jira/tomcat/apache-tomcat-8.5.42-atlassian-hosted
catalina.useNaming	true
com.atlassian.gadgets.dashboard.ignoreCache	true
com.sun.jndi.ldap.connect.pool.timeout	0
common.loader	"\${catalina.base}/lib"; "\${catalina.base}/lib/*"; "\${catalina.home}/lib"; "\${catalina.home}/lib/*"
file.encoding	UTF-8
file.encoding.pkg	sun.io
file.separator	/

## View the custom fields that take longest to index

When you experience a sudden degradation in indexing performance, it might be because a custom field takes long to index. Normally, re-indexing time is not evenly distributed and there are several fields which take up most of the indexing time.

To find out which custom fields might take longest to index, you can look up the metrics in the logs (available in Jira 8.10 and later) or click **Actions > Custom field indexing** for a specific node to view this data in the UI. The page displays the 20 most time-consuming custom fields (10 in the Total and 10 in the Snapshot time period).

The information which custom fields take up the majority of the indexing time allows you to take action to improve the performance. You might try changing the custom fields configuration or, if possible, improve the custom field indexer itself. If its a custom field thats not crucial to your business and its not a system custom field, you might try removing it in the test environment and see what indexing time gain you can achieve.

Jira
Search

---

**Administration** Search Jira admin

---

- Applications
- Projects
- Issues
- Manage apps
- User management
- Latest upgrade report
- System

### Custom field indexing

View the total and the snapshot metrics for the 10 custom fields that take the longest to index and affect indexing performance. Last data update on node node1: 12/Mar/20 3:15 PM. [Learn more](#)

[← Back to clustering](#)
[↻ Refresh](#)
[📄 Export to csv](#)

Time period	Custom field name	Custom field type	% of indexing time	Average indexing time (ms)	Max indexing time (ms)	Calls to indexer
Total	customfield_15200 (Rank which is so long that it takes more than one line and goes on the second line)	Custom	18.5%	16.1 ms	1 822 ms	2 717
Total	customfield_11002 (Sprint target)	3rd party app	7.8%	6.1 ms	1 778 ms	2 717
Total	watcher	System	7.2%	5.6 ms	1 466 ms	2 713
Total	customfield_23456 (Target delivery)	Custom	6%	4.7 ms	280 ms	2 715
Total	issuelinks	System	4.7%	3.7 ms	1 499 ms	2 715
Total	customfield_234567 (Date of release)	Custom	4.6%	3.6 ms	251 ms	2 715
Total	customfield_234 (EIS)	3rd party app	4.5%	3.5 ms	247 ms	2 711
Total	customfield_1234 (Development team)	3rd party app	2.3%	1.8 ms	373 ms	2 711
Total	assignee	System	2%	1.6 ms	1 367 ms	2 711
Total	customfield_2134 (Time on prod)	3rd party app	1.4%	1.1 ms	231 ms	2 712
Snapshot	epic	System	21.9%	24 ms	456 ms	1 340
Snapshot	customfield_9945 (Start of increment)	3rd party app	20.4%	12 ms	1 235 ms	1 340
Snapshot	labels	System	15.3%	13 ms	234 ms	1 340
Snapshot	customfield_76 (TTM)	3rd party app	12.5%	10 ms	456 ms	1 342
Snapshot	watcher	System	8.4%	7 ms	200 ms	1 340
Snapshot	customfield_276 (Private storage address)	3rd party app	8.3%	6 ms	100 ms	1 340
Snapshot	customfield_7777 (MAU)	Custom	8.1%	5.4 ms	77 ms	1 340
Snapshot	customfield_5567 (R&I)	Custom	7%	1.8 ms	100 ms	1 340
Snapshot	assignee	System	4.6%	1.1 ms	89 ms	1 340
Snapshot	customfield_1234 (Dev department)	3rd party app	1%	1 ms	70 ms	1 340

## Best practices for analyzing metrics

Its a good idea to refer to these metrics every time you introduce a configuration change or you make changes to the system in your test environment. If youre about to analyze the report, consider the following best practices:

To get reliable data, we recommend looking at the report after a full background re-index finished.

Its a good idea to analyze reports where there are more calls to indexer than the number of issues on an instance as this is the best quality data.

Before introducing any changes, look at the values in the **Max indexing time** column in the Snapshot section. The custom fields that score high there (for example reach 5000 milliseconds - 5 secs - or more) are the ones that tend to be most expensive index-wise.

## Analyze performance data

The table displays the custom fields that take longest to index or re-index (10 for Total and 10 for Snapshot). The data is displayed per node so it might be different depending on the node you select on the Clustering page. The data is refreshed in the background every 5 minutes. You can refer to the timestamp to see when the last indexing data update was sent by the system.

The data is sorted by indexing cost starting with the most time-consuming custom field in Total and in the Snapshot sections.

The table contains the following data:

**Time period:** sets the timeframe for the presentation of the data. Total is the time from the last full re-index or the start of Jira (if there was no full re-index since that time). Snapshot presents a 5-minute-timeframe (time from the last snapshot).

**Custom field name:** is the name of the custom field as used in JQL.

**Custom field type:** is the type of the custom field:

- **System** is a custom field used by the system. You cannot change its configuration or its indexing time.
- **3rd party app** is a custom field coming from a 3-rd party app you use. If it takes long to index, you might try changing its configuration or contact the vendor to improve the custom field indexer.
- **Custom** is a custom field that has been created on your instance. If it takes long to index, you might try changing its configuration or improve the custom field indexer.

**% of the indexing time:** is the % of time spent indexing or re-indexing a given custom field. It shows how a custom field affects indexing time.

**Max indexing time (ms):** is the maximum time spent indexing or re-indexing a given custom field.

**Average indexing time (ms):** is the average time spent indexing or re-indexing a given custom field. It's the sum in milliseconds of all the calls to indexer divided by the number of these calls.

**Calls to indexer:** is how many times the custom field indexer was called in a time period to re-index a given custom field.

#### Further analysis

To get more insights, you might also analyse the logs (available in Jira 8.1 and later):

Go to `atlassian/application-data/jira/log/atlassian-jira.log`

Use `grep indexing-stats` to find the data. It might look in the following way:

Regular log entry:

```
{field: epic, addIndex: {sum/allSum:38.5%, sum:1285825ms, avg:30.1ms, max:3822ms, count:42717}},
```

Re-indexing log entry:

```
{order:1,
 name:customfield_10000 (Approvals),
 isKnown:false,
 addIndex: {sum:2328373ms, avg:7.7ms, max:2750ms, count:302109},
 totalIndexTime:984864ms,
 addIndexSum/totalIndexTime:236.4%,
 numberOfIndexingThreads:20}
```

where:

`order` - fields that are indexed/re-indexed are displayed according to how long it took to index/re-index them

`sum/allSum` - is the % of time spent indexing a custom field vs all custom fields

`addIndex sum / avg / max / count` - is the the total time spent indexing a custom field / the average time spent indexing a custom field / maximum time spent indexing a custom field / calls to indexer

`totalIndexTime` - is the total indexing/re-indexing time

`addIndexSum/totalIndexTime` - is the cost (in time) of indexing/reindexing this field vs the total indexing/re-indexing time

`numberOfIndexingThreads` - number of indexing threads

For further explanation of the log stats, see [Indexing stats](#).

# Scheduler administration

The **Scheduler administration** page displays scheduled administrative jobs in Jira, their triggers, and properties.

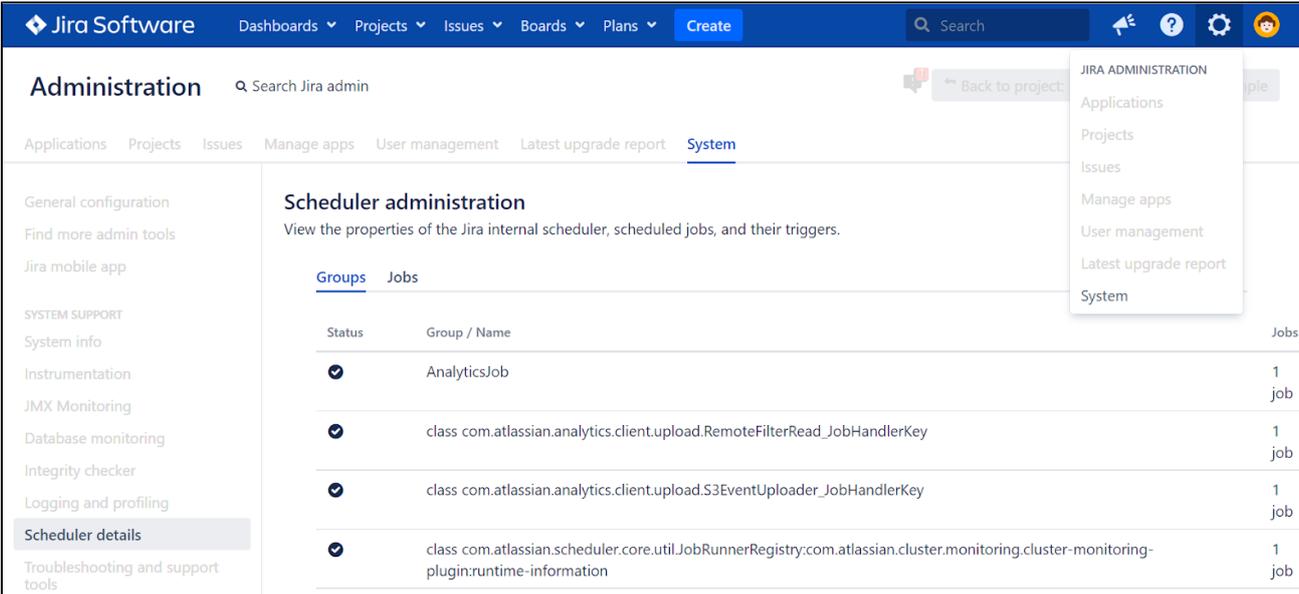
Scheduled jobs in Jira include repetitive tasks and events executed by cron at regular intervals. For example:

- cleanup tasks
- update tasks
- analytical tasks and events

## On this page:

- [Accessing Scheduler administration in Jira](#)
- [Editing and disabling scheduled jobs](#)
- [Related content](#)

The following image shows an example of the **Scheduler administration** page:



The screenshot shows the Jira Administration page for Scheduler administration. The page has a left sidebar with navigation options like 'General configuration', 'SYSTEM SUPPORT', and 'Scheduler details'. The main content area shows a table of scheduled jobs. The table has two tabs: 'Groups' and 'Jobs'. The 'Jobs' tab is selected, showing a list of jobs with their status (checked), group name, and the number of jobs (1 job).

Status	Group / Name	Jobs
✓	AnalyticsJob	1 job
✓	class com.atlassian.analytics.client.upload.RemoteFilterRead_JobHandlerKey	1 job
✓	class com.atlassian.analytics.client.upload.S3EventUploader_JobHandlerKey	1 job
✓	class com.atlassian.scheduler.core.util.JobRunnerRegistry:com.atlassian.cluster.monitoring.cluster-monitoring-plugin:runtime-information	1 job

The page contains two tabs **Groups** and **Jobs**:

- The **Groups** tab lists the packages of scheduled jobs. You can see the status of each group, how many jobs each group includes, and the schedule of each groups execution.
- The **Jobs** tab lists individual scheduled jobs. You can see the status of each job, the duration of each jobs last run, and the schedule of each jobs execution.

 You cant edit scheduled jobs on this page. But you can do it on the backend.

## Accessing Scheduler administration in Jira

You'll find the information on scheduled jobs on the **Scheduler administration** page.

To open the list of scheduled jobs:

1. Go to **Administration** () > **System**.
2. From the left-side menu, select **Scheduler details**.
3. To view the details of a group or job, select **Show more**.

 For more information on building cron expressions in Jira, see [Constructing cron expressions for a filter subscription](#).

## Editing and disabling scheduled jobs

Jira no longer allows editing and disabling scheduled jobs on the Scheduler administration page. But you can do it on the backend, through the Jira database.

To do it, you should have [System Admin access](#).

However, if you want to edit or disable any scheduled jobs, we recommend contacting [Atlassian Support](#) first.

For more information, see:

- [Jiras architecture, scheduled tasks, and background processes](#)
- [Jira database schemas](#)

## Related content

- [Error in Scheduler Administration "com.atlassian.jira.issue.subscription.DefaultSubscriptionManager: XXXX"](#)
- [The Jira Scheduler administration page is showing errors for the MailCleanerJobRunner and MailPullerJobRunner jobs](#)

# Configuring global settings

This section of the documentation contains information on how to check and configure settings in your Jira installation that are applied globally to all users. It includes information on default settings for your Jira installation, and default settings that apply to your users.

- [Configuring time tracking](#)
- [Configuring Jira application options](#)
  - [Configuring advanced settings](#)
  - [Configuring the Base URL](#)
  - [Configuring the administrator contact form](#)
- [Setting properties and options on startup](#)
  - [Recognized system properties for Jira applications](#)
- [Advanced Jira application configuration](#)
  - [Changing the constraints on historical time parameters in gadgets](#)
  - [Changing the default order for comments from ascending to descending](#)
  - [Limiting the number of issues returned from a search view such as an RSS feed](#)
- [Configuring file attachments](#)
- [Configuring issue linking](#)
- [Configuring issue cloning](#)
- [Configuring the allowlist](#)
- [Configuring sub-tasks](#)
- [Managing filters](#)
- [Managing dashboards](#)
- [Enabling logout confirmation](#)
- [Rich text editing](#)
- [Configuring terminology](#)

# Configuring time tracking

Jira's time tracking feature enables users to record the time they spend working on issues.

## Note:

- Before users can specify time estimates and log work, they must be granted the **Work On Issues** permission for the relevant project(s).
- For all of the following procedures, you must be logged in as a user with the **Jira Administrators** global permission.

## Disabling time tracking

Time tracking is ON by default (as shown in [screenshot 1 below](#)). However, this feature can be disabled from the Time Tracking administration page. **Time tracking will be OFF by default if your Jira installation was upgraded from a version prior to 4.2 that had time tracking either disabled or never enabled.**

1. Choose **Administration**  > **Issues**.
2. Select **Issue Features > Time Tracking** to open the Time Tracking page.
3. Click the '**Deactivate**' button to turn time tracking OFF.

**You will not lose any existing time tracking data by disabling/re-enabling time tracking.**

## Enabling time tracking

1. Choose **Administration**  > **Issues**.
2. Select **Issue Features > Time Tracking** to open the Time Tracking page.
3. Click the '**Activate**' button to turn time tracking ON.

### Screenshot 1: Time tracking is ON

**Time Tracking is currently ON.** 

**Note:** To change these values deactivate and then reactivate Time Tracking.

The number of working hours per day is **8**.  
The number of working days per week is **5**.  
Time estimates will be displayed in the following format: **pretty (e.g. 4 days, 4 hours, 30 minutes)**  
The current default unit for time tracking is **minute**.  
Copying of comments to work description is currently **enabled**.

For the users you wish to be able to log work on issues, ensure that they have the **Work On Issues** permission in the relevant [permission scheme](#).

To deactivate Time Tracking, simply click below.

## On this page:

- [Disabling time tracking](#)
- [Enabling time tracking](#)
- [Configuring time tracking settings](#)
- [Tracking time](#)
- [About 'Legacy Mode'](#)

**i** [Time tracking apps](#) for Jira in the Atlassian Marketplace extend Jira's time tracking power. [Check them out here.](#)

## Configuring time tracking settings

To edit Jira's time tracking settings, it must first be disabled. Once you have changed the settings, you will then need to re-enable time tracking so that users can log work on issues.

ⓘ You will not lose any existing time tracking data by disabling/re-enabling time tracking.

1. Choose **Administration** (⚙️) > **Issues**.
2. Select **Issue Features** > **Time Tracking** to open the Time Tracking page.
3. If Time Tracking is ON (refer to the indication at the top of the Time Tracking screen), click the **Deactivate** button to turn time tracking OFF.
4. The time tracking settings will now be editable as shown in the following screenshot.

5. Configure time tracking settings by editing the following fields:
  - 'Hours per day' enter a suitable value (e.g. 8). You can enter fractions if you wish.
  - 'Days per week' enter a suitable value (e.g. 5). You can enter fractions if you wish.
  - 'Time format' select **pretty/days/hours**. This will determine the format of the 'Time Spent' field when an issue is displayed.
  - 'Default Unit' select **minutes/hours/days/weeks**. This will be applied whenever your users log work on an issue without specifying a unit.
  - 'Legacy Mode' select this checkbox if you prefer to use Jira's time tracking features as they operated prior to Jira version 4.2. For more details about this option, please see [About 'Legacy Mode'](#) (below).
  - 'Copy Comment To Work Description' select this checkbox to ensure that any content entered into a Comment field while logging work as part of an issue operation, is also copied across to the Work Description.
    - ⓘ When 'Copy Comment To Work Description' is enabled, your user's work log entries will be visible only to members of the project role or group selected in the padlock icon drop-down on their issue operation screen. If 'Copy Comment To Work Description' is disabled, your user's work log entries will be visible to anyone by default.

6. Click the **Activate** button to turn time tracking ON.  
If the permission schemes used by your project(s) already have the appropriate Work On Issues permissions, then there is no need to proceed any further.  
However, if you need to configure these permissions, proceed with the remaining steps below:
7. Click the **permission scheme** link as shown in [screenshot 1 \(above\)](#). The 'Permissions Scheme' page will be displayed.
8. Click the **Permissions** link of the permission scheme associated with the project(s) where you wish to specify Work On Issues permissions. The 'Edit Permissions' page is displayed for your chosen permission scheme.

ⓘ See [Managing project permissions](#) for details about the various permissions.

9. Check whether the row labeled 'Work On Issues' contains the appropriate users, groups or project roles who need to specify time estimates or log work. If it does not, click the **Add** link in the 'Operations' column:

[Screenshot 3: Time tracking Permissions](#)

Time Tracking Permissions	Users / Groups / Project Roles	Operations
<b>Work On Issues</b> Ability to log work done against an issue. Only useful if Time Tracking is turned on.	• Project Role (Developers) (Delete)	<input type="button" value="Add"/>

10. Select the users, groups or project roles to whom you want to allow time tracking and work logging on issues.
11. Click the **Add** button.
12. If it is needed to enter the **Original Estimate** during issue creation or during issue editing, ensure that the field **Time Tracking** is added to the relevant screens associated with those operations. Refer [Associating a screen with an issue operation](#) for more details.

## Tracking time

You can either track time by entering the values in the Original estimate and the Remaining estimate fields on individual issues or when you edit multiple issues using the Bulk change option. Changing the values is also possible when you resolve issues through a bulk issue transition as long as time tracking is a part of the resolution screen.

Note that in single edit, when you enter only one of those values, the other one is automatically populated with the same value. For example, when you enter 4h as your Original estimate and leave the Remaining estimate empty, the Remaining estimate is automatically populated with the 4h value. In bulk edit this behaviour is there if you select to change both fields and leave one of them empty. If you select to change only one field, the other remains empty.

You can change both estimates as needed while you're working on issues unless you're in the Legacy Mode.

## About 'Legacy Mode'

- If Legacy Mode is disabled, your users will be able to change the **Original Estimate** value *irrespective of any work being logged on an issue*. Legacy Mode is disabled by default on new installations of Jira version 4.2 or later.
- If Legacy Mode is enabled, your users can only specify an **Original Estimate** *before* they start logging work on an issue. This value cannot be changed once *any* work has been logged, unless all work logs for that issue are first deleted. If not, you can only update the remaining estimate.
- By default,
  - **Legacy Mode** is disabled if your Jira 4.2 installation was conducted cleanly (that is, without upgrading from an earlier version of Jira).
  - **Legacy Mode** is enabled if you upgraded Jira from a version prior to 4.2.
- With Legacy Mode enabled, if you change the Remaining Estimate field in a workflow post function the Original Estimate is also cleared. This issue is tracked at

 [JRASERVER-25034](#) - Time Tracking Legacy Mode and Workflow Post Functions Error CLOSED



### Do more with Jira

#### Do more with Jira

Time tracking apps for Jira in the Atlassian Marketplace extend Jira's time tracking power. [Check them out.](#)

# Configuring Jira application options

Jira has a number of configuration options that allow your Jira application to be customized for use within your organization. These options can be accessed and edited on Jira's 'General Configuration' page.

## On this page:

- [Editing Jira's general configuration](#)
- [General settings](#)
- [Internationalization](#)
- [Options](#)

## Editing Jira's general configuration

1. In the upper-right corner of the screen, select **Administration**  **System**.
2. Select **General Configuration** to open the Administration page.
3. Click the **Edit Settings** button to edit the three sections as described below:
  - [Settings](#)
  - [Internationalization](#)
  - [Options](#)

 The Advanced Settings button is only visible if you have the **Jira System Administrators** global permission.

## General settings

(If marked with an \* the function is not available or editable in the Cloud)

Setting	Description
Title	This is the title that will be displayed on the Jira login page and the <a href="#">dashboard</a> . It helps identify your installation and its purpose.  Also see logo, which is displayed on every Jira page.
Mode *	Jira can operate in two modes: <ul style="list-style-type: none"><li>- <b>Public</b> Anyone, even people outside of your organization, can sign themselves up with self-registration and create issues (within the bounds of your Jira system's permissions).</li><li>- <b>Private</b> Useful for internal issue-tracking systems where you do not want public users to login. Self-signup is disabled; only Administrators can create new users.</li></ul>  If the Jira application has a LDAP directory configured with delegated authentication and the option <b>Copy user</b> is enabled, users will be able to login and create a new accounts. <i>Default: <b>Private</b></i>
Maximum Authentication Attempts Allowed *	The maximum authentication attempts that are allowed before CAPTCHA is shown to a user. If you leave it blank then CAPTCHA will never be shown and users will have unlimited authentication attempts. It is recommended that you set this to a small number (e.g. below 5). <i>Default: <b>3</b> (for new installations of Jira)</i>
CAPTCHA on signup *	If you are running Jira in Public mode (see above), it is strongly recommended that you enable CAPTCHA. This will show a CAPTCHA image on signup to prevent spambots from signing up. <i>Default: <b>ON</b></i>
Base URL *	The base URL of this Jira installation. You can only configure Jira to respond to a single URL and this setting <i>must</i> match the URL that your users request for accessing your Jira instance. You cannot (for example) have a different hostname or URL for internal and external users. This URL is also used in outgoing <a href="#">email notifications</a> as the prefix for links to Jira issues. Check out <a href="#">Configuring the Base URL</a> for more information.

Email from	<p>Specifies the <b>From:</b> header format in notification emails. Default is of the form "John Doe (Jira) &lt;jira@company.com&gt;". Available variables are '{\$fullname}', '{\$email}' and '{\$email.hostname}'. Note that the actual address (e.g. 'jira@company.com') cannot be specified here.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> The address is determined by the <a href="#">mail server</a> or individual <a href="#">project configuration</a>.</p> </div>
Introduction	<p>A short introduction message displayed on the <a href="#">dashboard</a>.  Also see the <a href="#">announcement banner</a>, which is displayed on every Jira page. You can include HTML, but ensure all tags are correctly closed.</p>

## Internationalization

Setting	Description
Indexing language	<p>Jira uses <a href="#">Lucene</a>, a high-performance text search engine library, in full-text searches for issues stored in Jira. This option is designed to enhance Jira's search indexing and issue searching features for issues entered in the languages available in this list. Hence, choose the language that matches the language used in your issues.</p> <p>Choosing a specific language in this list has the following effects when conducting searches in Jira (with respect to your chosen language):</p> <ul style="list-style-type: none"> <li>• Reserved words in text fields will not be indexed.</li> <li>• Stemming of words in all Jira fields will be active.</li> </ul> <p>If multiple languages are used in your issues (or you wish to disable the two effects above), choose <b>Other</b>.</p> <p> You will need to <a href="#">re-index</a> Jira if you change this value.</p>
Installed languages	<p>This section lists all language packs available within the Jira system. (Note: to install additional languages, see <a href="#">Internationalization</a>.)</p>
Default language	<p>The language used throughout the Jira interface (as selected from the list displayed in <b>Installed Languages</b> above). Users can override the default language by using the <b>Language</b> setting in their user profile.</p>
Default user time zone	<p>This is the time zone used throughout the Jira interface. Users can override the default time zone by using the <b>Time Zone</b> preference in their user profile. (To choose the time <i>format</i>, see <a href="#">Configuring the look and feel of your Jira applications</a>.)  Date fields that have no time component, such as due dates, release dates (associated with versions), and custom date fields, solely record date information (and no time zone-related information). These are not affected by time zone settings.</p>

## Options

Setting	Description
Allow users to vote on issues	<p>Controls whether voting is enabled in Jira. Voting allows users to indicate a preference for issues they would like to be completed or resolved. See also the 'View Voters and Watchers' <a href="#">permission</a>. For Jira Service Management, you can additionally enable voting for requests in the customer portal. For more info, see <a href="#">Managing access to your service project</a>.</p> <p><i>Default: <b>ON</b></i></p>

Allow users to watch issues	Controls whether watching is enabled in Jira. Users can 'watch' issues which they are interested in. Users watching an issue will be notified of all changes to it. See also the 'View Voters and Watchers' and 'Manage Watcher List' <a href="#">permissions</a> . <i>Default: <b>ON</b></i>
Allow users to share dashboards and filters with the public	Controls whether "Public" is an option for sharing a filter. Users can choose who to share a filter with, and "Public" allows them to share it with anonymous users i.e. user <i>not</i> logged into Jira. If your instance can be accessed publicly, these filters will be accessible by the general public. Changing the option to <b>OFF</b> will <b>not</b> change any existing filters that are shared as "Public". <i>Default: <b>ON</b></i>
Maximum project name size	Controls the maximum number of characters allowed for a project name. Changing this value will not affect the names of existing projects. <i>Default: <b>80</b></i>
Maximum project key size	Controls the maximum number of characters allowed for a project key. Changing this value will not affect the keys of existing projects. You can set this to any value between 2 and 255, inclusive. <i>Default: <b>10</b></i>
Allow unassigned issues	When turned <b>ON</b> , the default assignee for the project is <b>Unassigned</b> . When turned <b>OFF</b> , issues must always be assigned to someone - by default, the assignee will be the <b>Project Lead</b> as defined for each <a href="#">project</a> . <i>Default: <b>ON</b></i>
External user management	When turned <b>ON</b> , Jira will not display options for users to change their password and edit their profile. This will also disable the <b>Forgot your password</b> link on the login page. Generally you would only turn this <b>ON</b> if you are managing all your users from outside Jira (e.g. using <a href="#">Crowd</a> , <a href="#">Microsoft Active Directory</a> , or another LDAP directory) <i>Default: <b>OFF</b></i>
Logout confirmation	Controls whether to obtain user's confirmation when logging out: <b>NEVER COOKIE</b> - prompt for confirmation if the user was automatically logged in (via a cookie). <b>ALWAYS</b> <i>Default: <b>NEVER</b></i>
Use gzip compression	Controls whether to compress the web pages that Jira sends to the browser. It is recommended that this be turned ON, unless you are using mod_proxy. <i>Default: <b>ON</b></i>
User email visibility	Controls how users' email addresses are displayed in the user profile page. - <b>PUBLIC</b> - email addresses are visible to all. - <b>HIDDEN</b> - email addresses are hidden from all users. - <b>MASKED</b> - the email address is masked (e.g. 'user@example.com' is displayed as 'user at example dot com'). - <b>LOGGED IN USERS ONLY</b> - only users logged in to Jira can view the email addresses. <i>Default: <b>PUBLIC</b></i>
Comment visibility	Determines what will be contained in the list that is presented to users when specifying comment visibility and worklog visibility. - <b>Groups &amp; Project Roles</b> - the list will contain groups and project roles. - <b>Project Roles only</b> - the list will only contain project roles. <i>Default: <b>Project Roles only</b></i>
Exclude email header 'Precedence: bulk'	Controls whether to prevent the <b>Precedence: Bulk</b> header on Jira notification emails. This option should only be enabled when notifications go to a mailing list which rejects 'bulk' emails. In normal circumstances, this header prevents auto-replies (and hence potential mail loops). <i>Default: <b>OFF</b></i>
Issue Picker Auto-complete	Provides auto-completion of issue keys in the 'Issue Picker' popup screen. Turn OFF if your users' browsers are incompatible with AJAX. <i>Default: <b>ON</b></i>

JQL Auto-complete	Provides auto-completion of search terms when users perform an advanced (JQL) search. Turn OFF if you prefer not to use this feature, or are experiencing a performance impact. <i>Default: <b>ON</b></i>
Internet Explorer MIME Sniffing Security Hole Workaround Policy	Attachment viewing security options for cross-site site scripting vulnerabilities present in Internet Explorer 7 and earlier. Changes the default browser action for attachments in Jira. Options are: - <b>Insecure: inline display of attachments</b> - allows all attachments to be displayed inline. Only select this option if you fully understand the security risks. - <b>Secure: forced download of all attachments for all browsers</b> - force the download of all attachments. This is the most secure option, but is less convenient for users. - <b>Work around Internet Explorer security hole</b> - forced download of high-risk attachments (IE-only Workaround) - for IE browsers, force the download of attachments that IE would mistakenly detect as an HTML file. Declared HTML attachments are also never displayed inline. Use this option to reduce the risk of attacks to IE users via attachments. <i>Default: <b>Work around Internet Explorer security hole</b></i>
Contact Administrators Form	Provides an email form for users to fill in when they click the 'Contact Administrators' link (which appears when appropriate in Jira, e.g. on Login panels and pages). <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> Applies only if <a href="#">outgoing email</a> is enabled.</div> Can be used with or without the custom 'Contact Administrators Message' below. Users with the <b>Jira Administrators</b> <a href="#">global permission</a> (not <b>Jira System Administrators</b> - see <a href="#">JRA-27454</a> for details) will be notified as a result of this feature being used. To learn more on how to configure the form, see <a href="#">Configuring the administrator contact form</a> . <i>Default: <b>OFF</b></i>
Contact Administrators Message	Displays a custom message when users click the 'Contact Administrators' link (which appears when appropriate in Jira, e.g. on Login panels and pages). The 'Contact Administrators Message' will be displayed at the top of the 'Contact Administrators Form', only if the form is enabled (see above).
Allow Gravatars	Enables users to use <a href="#">Gravatars</a> in their user profile instead of Jira-specific avatars. Users will not be able to use Jira-specific avatars if Gravatars are enabled, and vice versa. <i>Default: <b>OFF</b></i>
Inline edit	Enables inline editing, i.e. click to edit a field on the screen. <i>Default: <b>ON</b></i>
Auto-update search results	Enables search results to be automatically updated when criteria are modified in a basic search. <i>Default: <b>ON</b></i>
Enable HTML in custom field descriptions and list item values	Allows to add HTML to the descriptions of custom fields and the values of list items. <i>Default: <b>OFF</b> (recommended for security)</i>

<p>Disable empty JQL queries</p>	<p>Allows you to choose how an empty JQL query behaves: either returning no results at all (ON), or all existing issues (OFF). We've introduced this option to avoid performance issues with empty filters returning all existing issues in your instance.</p> <p>Note that some apps may be using empty JQL queries to return all issues on purpose. Turning on this option will affect them. An empty JQL query in <b>Queues</b>, <b>SLAs</b> and <b>Reports</b> within Jira Service Management will continue to return all issues per project.</p> <p><i>Default: OFF</i></p>
<p>Enable extra options when exporting to CSV</p>	<p>Allows you to choose a delimiter when exporting your search results to a CSV file. When exporting, you can choose to separate the values with one of these delimiters:</p> <ul style="list-style-type: none"> <li>• Comma (,)</li> <li>• Semicolon (;)</li> <li>• Vertical bar ( )</li> <li>• Caret (^)</li> </ul> <p>If you turn off this option, comma (,) will be used as the default delimiter. It's useful if you're making a lot of exports and don't want the extra dialog to be displayed every time.</p> <p><i>Default: ON</i></p>
<p>Max timeout of Favorite Filters gadget</p>	<p>The Favorite Filters gadget shows a list of favorite filters and the number of issues each of them contains. This option allows you to set the max timeout after which Jira stops counting the issues and instead displays the "Too many issues" message next to the filter. If you have large filters, counting the issues takes time and might affect Jira performance, so use this option to avoid that.</p> <p><i>Default: 5000 (ms)</i></p> <p>If you set it to -1, Jira wont be counting issues at all.</p>
<p><b>Display URL parameters in security dialogs (8.12.1+)</b></p>	<p>If disabled, the setting prevents URL parameter values from being captured and displayed on security dialogs such as the session or XSFR token expiry dialogs. This is to prevent possible phishing attacks and tackle security vulnerabilities.</p> <p><i>Default: OFF</i></p>

# Configuring advanced settings

Jira has a small number of commonly edited advanced configuration options, which are stored in the Jira database. These options can be accessed and edited from the **Advanced Settings** page. You must be a [Jira System Administrator](#) to do this.

## Editing Jira's advanced settings

### To access and edit options on the 'Advanced Settings' page:

1. In the upper-right corner of the screen, select **Administration**  **System**.
  2. Click the **Advanced Settings** button on the 'General Configuration' page.
  3. Edit the value of a **Key** by clicking its value on the right of the page and modifying the existing value.
- The table below has extended information on some of the **Key** values.

Key	Key configuration
jira.attachments.number.of.zip.entries	<a href="#">Configuring the number of files shown in the content of ZIP-format files on issues</a>
jira.clone.prefix	<a href="#">Configuring the cloned issue summary field prefix</a>
jira.date.picker.java.format	<a href="#">Configuring date picker formats</a>
jira.date.picker.javascrip.format	
jira.date.time.picker.java.format	
jira.date.time.picker.javascrip.format	

jira. issue. actions. order	<a href="#">Changing the default order for comments from ascending to descending</a>
jira. projectkey. pattern	<a href="#">Changing the Project Key Format</a>
jira. search. views. default. max	The default maximum number of issues exported from search results. Users can override the default by changing the URL <code>tempMax</code> parameter. This value must always be lower than or equal to the <code>jira.serach.views.max.limit</code> value.
jira. search. views. .max. limit	The absolute limit on the number of issues that can be exported from search results.
jira. table. cols. subtasks	<a href="#">Configuring sub-task fields displayed on parent issues</a>
jira. view. issue. links. sort. order	<a href="#">Configuring the order of linked issues displayed on the 'view issue' page</a>
jira. text. field. character. limit	This property limits the number of characters that can be entered into <b>Description, Environments, Comments</b> and <b>text custom fields</b> . The maximum is 2147483647. A value of 0 means unlimited characters.
jira. comment. collapsing. minimum. hidden	The minimum number of comments needed before the comment collapsing is enabled.
jira. newsletter. tip. delay. .days	The number of days before a prompt to sign up to the Jira applications insiders newsletter is shown. A value of -1 disables this functionality.

jira.bulk.create.max.issues.per.import	This property allows you to set the maximum number of issues a user can import via CSV at one time. The maximum is 2147483647. Entering a value of 0 will disable the importer for users.
jira.export.html.enabled	Specifies whether users can export the JQL search results to HTML.
jira.quicksearch.max.concurrent.searches	<p>The maximum number of concurrent searches that your users can perform by using the quick search. The limit applies to a single Jira instance (if you have Data Center with 5 nodes, the limit increases fivefold.)</p> <p>Many concurrent searches will affect Jira's performance. You can use <a href="#">JMX monitoring</a> to see how your users are searching, and determine the best limit.</p>
crowd.encrypted.encrypted.default	<p>This property defines password encryption algorithms:</p> <ul style="list-style-type: none"> <li>• <code>DISABLED</code> - disable the encryption. Its an insecure option. We dont recommend using it.</li> <li>• <code>BASE64</code> - obfuscates passwords by encoding them with Base64. Its an insecure option but it can be helpful in some cases. This option doesnt require encryption keys. Learn more about backing up data.</li> <li>• <code>AES_CBC_PKCS5Padding</code> - standard AES implementation in CBC mode. We recommend using this option.</li> </ul>
com.atlassian.jira.issue.fields.update.init.required	<p>This property is related to the custom field usage data that appears on the Custom fields page (the <i>Last value update</i> column). It lets you recalculate this data based on the information from the database, which is useful if the data is corrupt and you need to recalculate it from scratch. When you set the property to true, a one-time recalculation will be performed during the next run of the <code>CustomFieldUsageRecalculationJob</code>. You can check the details of this job by going to Scheduler details in the administration area. Once the recalculation is successful, the property is set back to false.</p>

- Click the **Update** button (which will appear in the **Operations** column on the right) to save the new value in the Jira database.

**Please Note:**

- Any changes you make to these properties/keys become effective immediately.
- Click the **General Settings** button to return to the **General Configuration** page.

## Related information

There are a handful of other advanced configuration options (which are of little interest to most Jira system administrators) whose default values can be customized in the `jira-config.properties` file located in the [Jira application home directory](#), which you may want to edit. For details, please see [Advanced Jira configuration](#).

# Configuring the Base URL

The **Base URL** is the URL via which users access Jira applications. The base URL **must** be set to the same URL by which browsers will be viewing your Jira instance.

Jira will automatically detect the base URL during setup, but you may need to set it manually if your site's URL changes or if you set up Jira from a different URL to the one that will be used to access it publicly.

You need to have the **system administrator global permission** in order to perform this function.

## To configure the Base URL:

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. Choose **General Configuration** in the left-hand panel.
3. Choose **Edit Settings**.
4. Enter the new URL in the **Base URL** text box.
5. Choose **Save**.

## Example

If Jira is installed to run in a non-root context path (that is, it has a context path), then the server base URL should include this context path. For example, if Jira is running at:

```
http://www.foobar.com/Jira
```

then the server base URL should be:

```
http://www.foobar.com/Jira
```

## Notes

- **Using different URLs.** If you configure a different base URL or if visitors use some other URL to access Jira, it is possible that you may encounter errors while viewing some pages.
- **Changing the context path.** If you change the context path of your base URL, you may also need to edit the web server's `server.xml` file to reflect the new path:
  1. Stop the Jira server.
  2. Go to your Jira 'destination directory'. This is the directory where the Confluence installation files are stored. For example, `C:\Program Files\Atlassian\JIRA`. Let's call this directory '{Jira\_INSTALLATION}'.
  3. Edit the configuration file at `{Jira_INSTALLATION}\conf\server.xml`.
  4. Change the value of the `path` attribute in the `Context` element to reflect the context path. For example, if Jira is running at `http://www.foobar.com/Jira`, then your `path` attribute should look like this:

```
<context path="/JIRA" docBase="../../JIRA" debug="0" reloadable="false" useHttpOnly="true">
```

5. Save the file.
- **Proxies.** If you are running behind a proxy, ensure that the proxy name matches the base URL. For example: `proxyName="foobar.com" proxyPort="443" scheme="https"`. This will make sure we are passing the information correctly.
  - This information needs to be added in the `Connector` element at `{Jira_INSTALLATION}\conf\server.xml`.

# Configuring the administrator contact form

The administrator contact form allows a user to send a message to the administrators of their Jira site by clicking the 'Contact Administrators' link. For example, the link appears on Login panels and pages.

## Customizing the administrator contact message

You can customize the message that is presented to the user on the '**Contact Administrators Form**' .

To edit the administrator contact message:

1. Choose the **Administration**  > **System** .
2. Choose **General Configuration**.
3. Click **Edit Settings**.
4. Scroll down to the **Contact Administrators Form** and set it to ON.
5. Enter your text in the **Contact Administrators Message** box. If you need markup assistance, click the ? under the box.
6. Click **Update**.

 If users send a message with the contact form it will reach all admins in the Jira admin group.

## The Default Administrator Contact Message

By default, the contact administrators message looks the following:

### Contact Site Administrators

---

To

From

Subject

Request details

---

To restore the message to its default simply remove the custom message you entered so that the Contact Administrators Message field is empty.

### Disabling the Contact Administrators Form

To enable or disable the administrator contact form:

1. Choose the **Administration**  > **System**.
2. Choose **General Configuration**.
3. Click **Edit Settings**.
4. Scroll down to the **Contact Administrators Form** and set it to OFF.
5. Click **Update**.

# Setting properties and options on startup

This page describes how to set Java properties and options on startup for Jira.

## On this page:

- [Linux](#)
- [Windows \(starting from .bat file\)](#)
- [Windows service](#)
- [Verifying your settings](#)
- [List of startup parameters](#)
- [Auditing properties](#)
- [Custom fields](#)

## Linux

### To Configure System Properties in Linux Installations,

1. From <jira-install>/bin, open **setenv.sh**.
2. Find the section **JVM\_SUPPORT\_RECOMMENDED\_ARGS=**
3. Refer to the list of parameters [below](#).

 Add all parameters in a space-separated list, inside the quotations.

## Windows (starting from .bat file)

### To Configure System Properties in Windows Installations When Starting from the .bat File,

1. From <jira-install>/bin, open **setenv.bat**.
2. Find the section **set JVM\_SUPPORT\_RECOMMENDED\_ARGS=**
3. Refer to the list of parameters [below](#).

 Add all parameters in a space-separated list, inside the quotations.

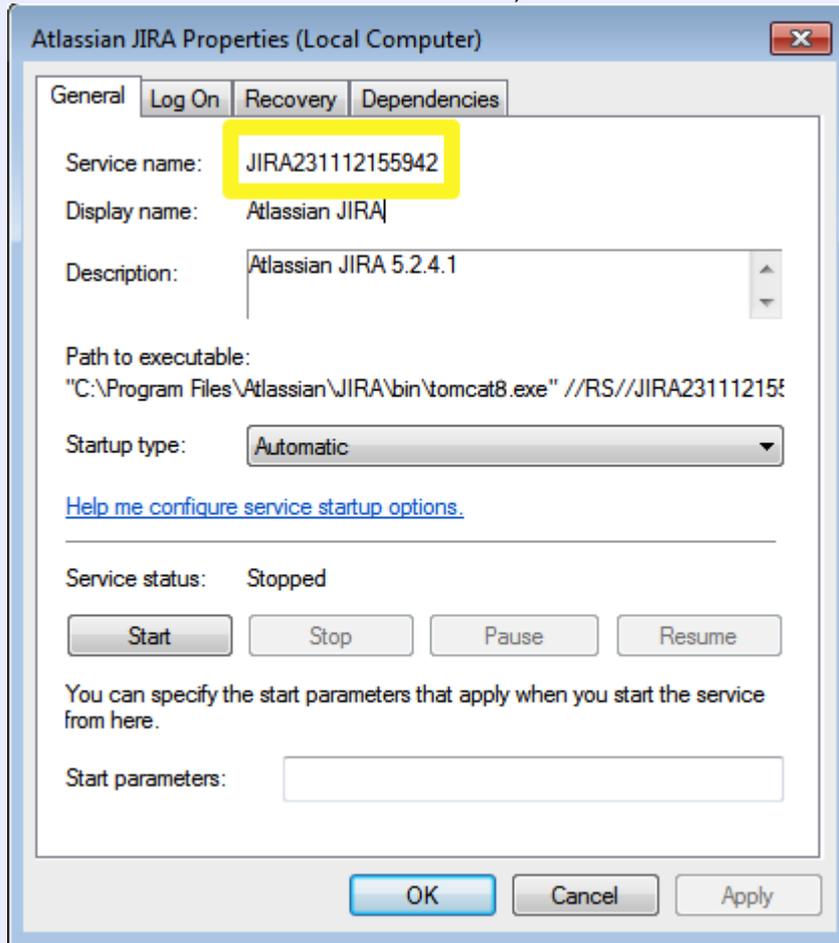
## Windows service

There are two ways to configure system properties when starting [Running Jira as a Windows service](#), either via [command line](#) or [in the Windows registry](#).

## Setting properties for Windows services via command line

### Setting Properties for Windows Services via Command Line

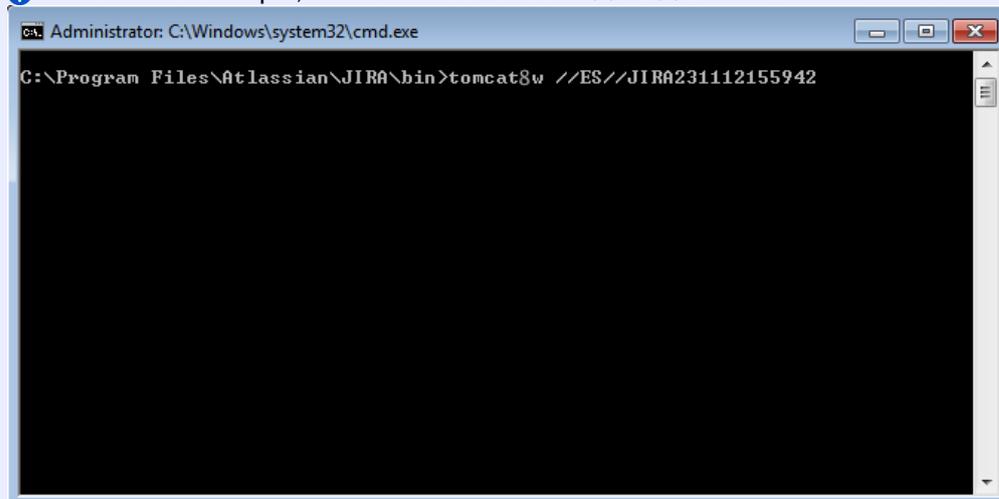
1. Identify the name of the service that Jira is installed as in Windows ( Control Panel > Administrative Tools > Services ):



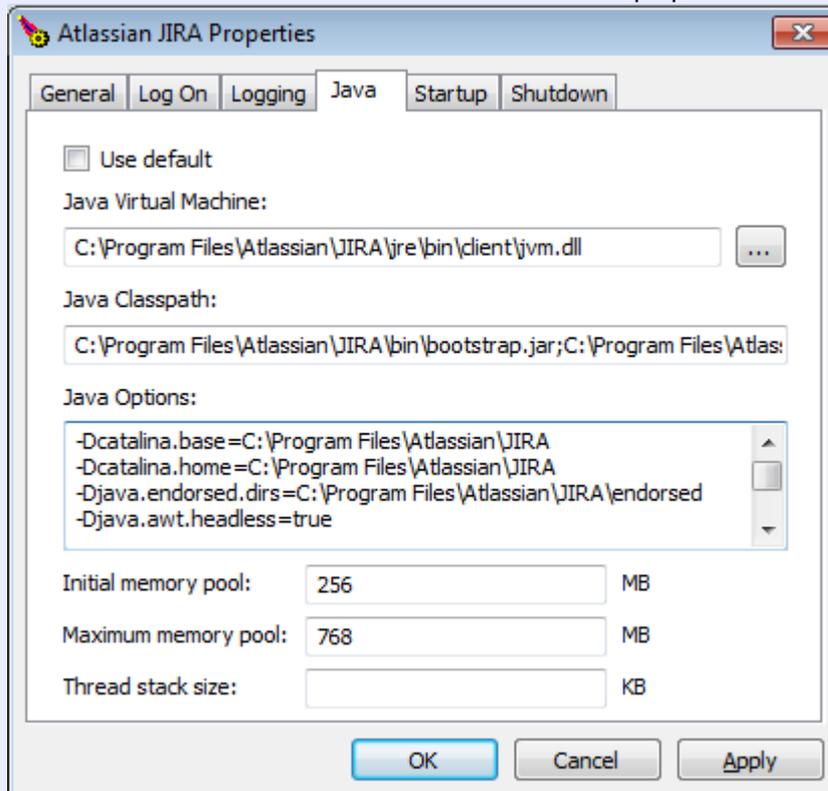
- i** In the above example, the **SERVICENAME** is: JIRA231112155942
2. Open the command window from Start >> Run >> type in 'cmd' >> Enter
3. cd to the bin directory of your Jira application installation directory.
4. Run:

```
tomcat8w //ES//%SERVICENAME%
```

- i** In the above example, it would be tomcat8w //ES//JIRA231112155942



5. Click on the `Java` tab to see the list of current start-up options:



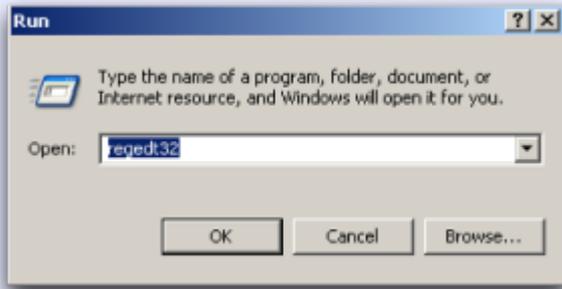
6. Append any new option on its own new line by adding to the end of the existing Java options. Refer to the list of parameters [below](#).

### Setting properties for Windows services via the Windows registry

In some versions of Windows, there is no option to add Java variables to the service. In these cases, you must add the properties by viewing the option list in the registry.

### To Set Properties for Windows Services via the Windows Registry,

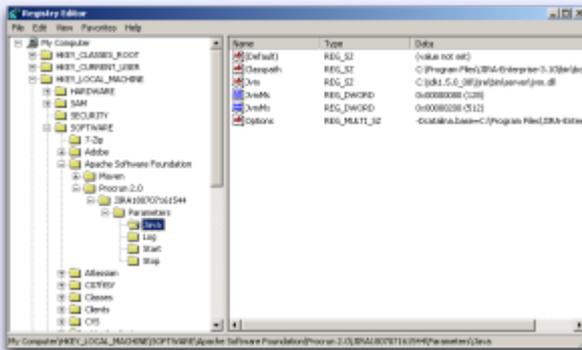
1. Go to Start >> Run, and run "regedit32.exe".



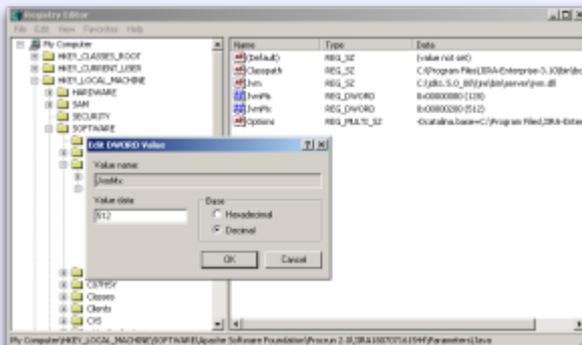
2. Find the Services entry:

**32-bit:** HKEY\_LOCAL\_MACHINE >> SOFTWARE >> Apache Software Foundation >> Procrun 2.0 >> JIRA

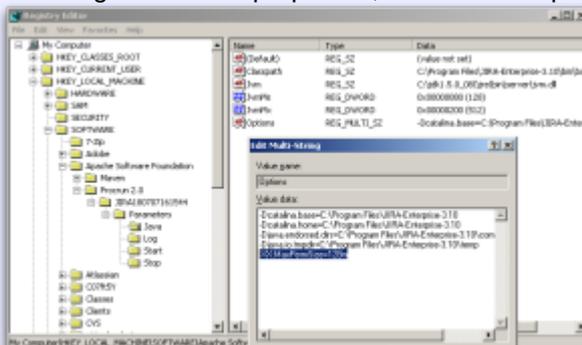
**64-bit:** HKEY\_LOCAL\_MACHINE >> SOFTWARE >> Wow6432Node >> Apache Software Foundation >> Procrun 2.0 >> JIRA



3. To change existing properties, especially increasing Xmx memory, double-click the appropriate value.



4. To change additional properties, double-click options.



5. Refer to the list of parameters [below](#). Enter each on a separate line.

### Verifying your settings

To verify what settings are in place, check the `<jira-home>/logs/atlassian-jira.log` or `catalina.out` file. A section in the startup appears like this:

```
JVM Input Arguments : -Djava.util.logging.config.file=/usr/local/jira/conf/logging.properties -XX:
MaxPermSize=256m -Xms256m -Xmx384m -Djava.awt.headless=true -Datlassian.standalone=JIRA -Dorg.apache.
jasper.runtime.BodyContentImpl.LIMIT_BUFFER=true -Dmail.mime.decodeparameters=true -Djava.util.logging.
manager=org.apache.juli.ClassLoaderLogManager -Djava.endorsed.dirs=/usr/local/jira/endorsed -Dcatalina.
base=/usr/local/jira -Dcatalina.home=/usr/local/jira -Djava.io.tmpdir=/usr/local/jira/temp
```

This display is also available by [viewing your system information](#).

## List of startup parameters

Memory property	Notes	Related pages
-Xmx -Xms XX:MaxPermSize	These properties are pre-existing. See related pages for instructions.	<a href="#">Increasing Jira memory</a>
-XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -XX:+PrintGCCause  -Xloggc:C:\Program Files\Atlassian\Application Data\JIRA\log\atlassian-jira-gc-%t.log  -XX: +UseGCLogFileRotation  -XX: NumberOfGCLogFiles=5  -XX:GCLogFileSize=20M	These properties are pre-existing, and are used for Garbage Collection tuning.	<a href="#">Using Garbage Collection Logs to Analyze Jira Application Performance</a> <a href="#">Analyze OutofMemory errors in Jira server with Heap Dumps</a>
-agentlib: yjpagent=onexit=memory, dir=/path/to/write /snapshots		<a href="#">[Archived] Profiling Memory and CPU Usage with YourKit</a>
-XX: InitialCodeCacheSize=32m  -XX: ReservedCodeCacheSize =512m	These properties are pre-existing, and are used to configure the size of the JVM code cache. A high value of reserved size allows Jira to load more installed apps.  The default configuration should be optimal for most Jira instances and solve any problems with the code cache getting full.	<a href="#">KB article: Jira crashes due to the CodeCache</a>
Mail property	Notes	Related pages

-Datlassian.mail.senddisabled -Datlassian.mail.fetchdisabled -Datlassian.mail.popdisabled	Set to 'true' to disable mail. In Linuxsetenv.sh, there is a pre-existing flag to uncomment.	<a href="#">Migrating Jira to another server</a> <a href="#">Notifications Are Issued for Incorrect Jira Issues</a>
-Dmail.debug	If set to "true", logs statements related to mail	<a href="#">Configuring Jira's SMTP mail server to send notifications</a> <a href="#">Creating issues and comments from email</a>
-Dmail.mime.decodestrict		<a href="#">Unable to Decode Mail Subject or Body when Creating Issue From Email</a>
-Dmail.imap.auth.disable -Dmail.imaps.auth.disable		<a href="#">IMAP setup fails with AUTHENTICATE Failed error in logs in Jira server</a>
-Dmail.imap.starttls.enable		<a href="#">Jira server unable to retrieve messages from IMAP server with No login methods supported error</a>
-Dmail.mime.decodeparameters	Sets mail handler to work correctly with emails from RFC 2231-compliant mail clients.	
-Dmail.smtp.localhost		<a href="#">Problems Sending Email from Jira - EHLO requires domain address</a>
<b>Encoding property</b>	<b>Notes</b>	<b>Related pages</b>
-Dfile.encoding -Dsun.jnu.encoding	Set to utf-8 for encoding consistency	<a href="#">Characters Not Supported by ASCII are Being Displayed as Question Marks</a> <a href="#">Internationalization and Encoding Troubleshooting</a> <a href="#">SQL Exception while updating issues or importing data in Jira applications with MySQL due to encoding</a> <a href="#">International Characters in Notification Email Subject Lines Are Being Replaced with Question Mark</a>
<b>Other Properties</b>	<b>Notes</b>	<b>Related pages</b>
-Duser.timezone		<a href="#">Incorrect Times Displayed in Jira</a>

-Dsvnkit.http.methods	Values includeBasic,Digest,Negotiate,NTLM	<a href="#">Jira Startup Fails Due to 'java.lang.SecurityException Unable to locate a login configuration' Subversion Plugin Displays 'An unknown error occurred - actions == null' Due to SVN Authentication</a>
-Dorg.apache.jasper.runtime.BodyContentImpl.LIMIT_BUFFER	true	<a href="#">(Archived) OutOfMemory Due to Tomcat Memory Leak JRA-10145</a>
-ea/-da	Enable/Disable assertions	<a href="#">java.lang.AssertionError When Sending Mail Via SMTP</a>
-Djava.net.preferIPv4Stack		<a href="#">SocketException to Announce 'Invalid argument' for an Available Port</a>
-Djavax.net.ssl.trustStore		<a href="#">Unable to Send Email 'javax.net.ssl.SSLException' Due to SMTP Server via SSL</a>
-Djava.awt.headless	Ships with true by default. Allows thumbnail generation.	
-Dhttp.proxyHost -Dhttp.proxyPort -Dhttps.proxyHost -Dhttps.proxyPort	Outbound Proxy Server hostname and port	<a href="#">How to Configure an Outbound HTTP and HTTPS Proxy for Jira applications</a>
-Dorg.apache.catalina.SESSION_COOKIE_NAME		<a href="#">Logging into another Atlassian application logs me out of Confluence</a>
-Datlassian.plugins.enable.wait	Time Jira waits for apps to load.	<a href="#">Jira System Plugin Timeout While Waiting for Plugins to Enable</a>
-Datlassian.plugins.startup.options="--disable-all-addons --disable-addons=com.atlassian.test.plugin"	Allows Jira to start with all user installed or specific user installed apps disabled. For more information on manual start up and specifying apps, see <a href="#">Start and Stop Jira applications</a> .	<a href="#">Start and Stop Jira applications</a>
-Dhide.system.error.details	Hides the details of errors that are displayed after starting Jira. The page (johnson) where these are displayed will still notify you about the errors.	
-Djira.startup.warnings.disable	Disables the page (johnson) that displays errors after starting Jira if there are only dismissible warnings. The page will appear if there are any important errors.	

<code>-Dcom.atlassian.streams.internal.LocalActivityProviders.allowed.wallclock.percentage</code>		
	Specifies the percentage of CPU time assigned to Activity Stream requests. Set the value to to an integer from 0 to 100. The default value is 10.	<a href="#">Jira Core 8.19.x upgrade notes</a>

## Auditing properties

These properties control the auditing feature, determining the number of audit entries logged, or stored in the database, and the size of those entries. Changing these settings will only affect new audit entries.

Increasing the amount of auditing done may have an adverse effect on performance.

Default value	Description
<code>plugin.audit.search.max.concurrent.nontext.requests</code>	
10	Maximum number of concurrent non-freetext search requests allowed, defaults to 10 per node
<code>plugin.audit.search.max.concurrent.text.requests</code>	
5	Maximum number of concurrent freetext search requests allowed, defaults to 5 per node
<code>plugin.audit.search.query.timeout</code>	
30	Timeout in seconds for a queued search request, defaults to 30 seconds
<code>plugin.audit.db.limit.rows</code>	
1000000	Maximum number of audit event rows stored in DB, events exceeding the limit get deleted in time order, defaults to 10M checked on hourly basis
<code>plugin.audit.db.limit.buffer.rows</code>	
1000	Buffer to accommodate new audit events, defaults to 1000 rows
<code>plugin.audit.db.delete.batch.limit</code>	
10000	maximum number of events to be deleted per database transaction used when enforcing retention limits, defaults to 10,000 rows
<code>plugin.audit.schedule.db.limiter.interval.mins</code>	
60	Database size check, running every 60 minutes
<code>plugin.audit.broker.exception.loggedCount</code>	
3	Maximum number of audit events written to system log file in case of error, defaults to 3
<code>plugin.audit.retention.interval.hours</code>	
24	Database retention check, which deletes events exceeding retention period, running every 24 hours
<code>plugin.audit.file.max.file.size</code>	
100	Size limit in megabytes for individual audit file, file rotates when limit is reached, defaults to 100MB
<code>plugin.audit.file.max.file.count</code>	

100	Maximum number of audit files, the earliest file will be deleted when limit is reached, defaults to 100
<code>plugin.audit.consumer.buffer.size</code>	
10000	Maximum number of audit events kept in buffer waiting to be consumed, defaults to 10,000
<code>plugin.audit.broker.default.batch.size</code>	
3000	Maximum number of audit events dispatched to consumer, defaults to 3,000 per batch
<code>plugin.audit.coverage.cache.read.expiration.seconds</code>	
30	How long the coverage cache is valid, defaults to 30 seconds

## Custom fields

Default value	Description
<code>jira.custom.field.indexing.batch.size</code>	
50	<p>To decrease reindex time and improve overall performance, custom field values are fetched in batches of 50. This speeds up time-consuming operations such as database search. Currently the property works only with Jira built-in custom fields.</p> <p>To disable the property, change the default to 1. Changing the property values, and getting the property to work for the first time requires full reindex.</p>

# Recognized system properties for Jira applications

Jira supports some configuration and debugging settings that can be enabled through Java system properties. System properties are usually set by passing the `-D` flag to the Java virtual machine in which Jira is running. See [Setting properties and options on startup](#).

## List of startup parameters

Memory property	Notes	Related pages
-Xmx -Xms XX:MaxPermSize	These properties are pre-existing. See related pages for instructions.	<a href="#">Increasing Jira memory</a>
-XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -XX:+PrintGCCause -Xloggc:C:\Program Files\Atlassian\Application Data\JIRA\log\atlassian-jira-gc-%t.log -XX: +UseGCLogFileRotation -XX: NumberOfGCLogFiles=5 -XX:GCLogFileSize=20M	These properties are pre-existing, and are used for Garbage Collection tuning.	<a href="#">Using Garbage Collection Logs to Analyze Jira Application Performance</a> <a href="#">Analyze OutofMemory errors in Jira server with Heap Dumps</a>
-agentlib: yjpagent=onexit=memory, dir=/path/to/write /snapshots		<a href="#">[Archived] Profiling Memory and CPU Usage with YourKit</a>
-XX: InitialCodeCacheSize=32m -XX: ReservedCodeCacheSize=512m	These properties are pre-existing, and are used to configure the size of the JVM code cache. A high value of reserved size allows Jira to load more installed apps.  The default configuration should be optimal for most Jira instances and solve any problems with the code cache getting full.	<a href="#">KB article: Jira crashes due to the CodeCache</a>
Mail property	Notes	Related pages
-Datlassian.mail. senddisabled -Datlassian.mail. fetchdisabled -Datlassian.mail. popdisabled	Set to 'true' to disable mail. In Linuxsetenv.sh, there is a pre-existing flag to uncomment.	<a href="#">Migrating Jira to another server</a> <a href="#">Notifications Are Issued for Incorrect Jira Issues</a>
-Dmail.debug	If set to "true", logs statements related to mail	<a href="#">Configuring Jira's SMTP mail server to send notifications</a> <a href="#">Creating issues and comments from email</a>

-Dmail.mime.decodeplaintext.strict		Unable to Decode Mail Subject or Body when Creating Issue From Email
-Dmail.imap.auth.plain.disable -Dmail.imaps.auth.plain.disable		IMAP setup fails with AUTHENTICATE Failed error in logs in Jira server
-Dmail.imap.starttls.enable		Jira server unable to retrieve messages from IMAP server with No login methods supported error
-Dmail.mime.decodeparameters	Sets mail handler to work correctly with emails from RFC 2231-compliant mail clients.	
-Dmail.smtp.localhost		Problems Sending Email from Jira - EHLO requires domain address
<b>Encoding property</b>	<b>Notes</b>	<b>Related pages</b>
-Dfile.encoding -Dsun.jnu.encoding	Set to utf-8 for encoding consistency	<a href="#">Characters Not Supported by ASCII are Being Displayed as Question Marks</a> <a href="#">Internationalization and Encoding Troubleshooting</a> <a href="#">SQL Exception while updating issues or importing data in Jira applications with MySQL due to encoding</a> <a href="#">International Characters in Notification Email Subject Lines Are Being Replaced with Question Mark</a>
<b>Other Properties</b>	<b>Notes</b>	<b>Related pages</b>
-Duser.timezone		<a href="#">Incorrect Times Displayed in Jira</a>
-Dsvnkit.http.methods	Values includeBasic,Digest,Negotiate,NTLM	<a href="#">Jira Startup Fails Due to 'java.lang.SecurityException Unable to locate a login configuration'</a> <a href="#">Subversion Plugin Displays 'An unknown error occurred - actions == null' Due to SVN Authentication</a>
-Dorg.apache.jasper.runtime.BodyContentImpl.LIMIT_BUFFER	true	<a href="#">(Archived) OutOfMemory Due to Tomcat Memory Leak</a> JRA-10145
-ea/-da	Enable/Disable assertions	<a href="#">java.lang.AssertionError When Sending Mail Via SMTP</a>

-Djava.net.preferIPv4Stack		<a href="#">SocketException to Announce 'Invalid argument' for an Available Port</a>
-Djavax.net.ssl.trustStore		<a href="#">Unable to Send Email 'javax.net.ssl.SSLEnvironmentException' Due to SMTP Server via SSL</a>
-Djava.awt.headless	Ships with true by default. Allows thumbnail generation.	
-Dhttp.proxyHost -Dhttp.proxyPort -Dhttps.proxyHost -Dhttps.proxyPort	Outbound Proxy Server hostname and port	<a href="#">How to Configure an Outbound HTTP and HTTPS Proxy for Jira applications</a>
-Dorg.apache.catalina.SESSION_COOKIE_NAME		<a href="#">Logging into another Atlassian application logs me out of Confluence</a>
-Datlassian.plugins.enable.wait	Time Jira waits for apps to load.	<a href="#">Jira System Plugin Timeout While Waiting for Plugins to Enable</a>
-Datlassian.plugins.startup.options="--disable-all-addons --disable-addons=com.atlassian.test.plugin"	Allows Jira to start with all user installed or specific user installed apps disabled. For more information on manual start up and specifying apps, see <a href="#">Start and Stop Jira applications</a> .	<a href="#">Start and Stop Jira applications</a>
-Dhide.system.error.details	Hides the details of errors that are displayed after starting Jira. The page (johnson) where these are displayed will still notify you about the errors.	
-Djira.startup.warnings.disable	Disables the page (johnson) that displays errors after starting Jira if there are only dismissible warnings. The page will appear if there are any important errors.	
-Dcom.atlassian.streams.internal.LocalActivityProviders.allowed.wallclock.percentage		
	Specifies the percentage of CPU time assigned to Activity Stream requests. Set the value to an integer from 0 to 100. The default value is 10.	<a href="#">Jira Core 8.19.x upgrade notes</a>

## Auditing properties

These properties control the auditing feature, determining the number of audit entries logged, or stored in the database, and the size of those entries. Changing these settings will only affect new audit entries.

Increasing the amount of auditing done may have an adverse effect on performance.

Default value	Description
plugin.audit.search.max.concurrent.nontext.requests	

10	Maximum number of concurrent non-freetext search requests allowed, defaults to 10 per node
plugin.audit.search.max.concurrent.text.requests	
5	Maximum number of concurrent freetext search requests allowed, defaults to 5 per node
plugin.audit.search.query.timeout	
30	Timeout in seconds for a queued search request, defaults to 30 seconds
plugin.audit.db.limit.rows	
10000000	Maximum number of audit event rows stored in DB, events exceeding the limit get deleted in time order, defaults to 10M checked on hourly basis
plugin.audit.db.limit.buffer.rows	
1000	Buffer to accommodate new audit events, defaults to 1000 rows
plugin.audit.db.delete.batch.limit	
10000	maximum number of events to be deleted per database transaction used when enforcing retention limits, defaults to 10,000 rows
plugin.audit.schedule.db.limiter.interval.mins	
60	Database size check, running every 60 minutes
plugin.audit.broker.exception.loggedCount	
3	Maximum number of audit events written to system log file in case of error, defaults to 3
plugin.audit.retention.interval.hours	
24	Database retention check, which deletes events exceeding retention period, running every 24 hours
plugin.audit.file.max.file.size	
100	Size limit in megabytes for individual audit file, file rotates when limit is reached, defaults to 100MB
plugin.audit.file.max.file.count	
100	Maximum number of audit files, the earliest file will be deleted when limit is reached, defaults to 100
plugin.audit.consumer.buffer.size	
10000	Maximum number of audit events kept in buffer waiting to be consumed, defaults to 10,000
plugin.audit.broker.default.batch.size	
3000	Maximum number of audit events dispatched to consumer, defaults to 3,000 per batch
plugin.audit.coverage.cache.read.expiration.seconds	
30	How long the coverage cache is valid, defaults to 30 seconds

### Custom fields

Default value	Description
jira.custom.field.indexing.batch.size	

50	<p>To decrease reindex time and improve overall performance, custom field values are fetched in batches of 50. This speeds up time-consuming operations such as database search. Currently the property works only with Jira built-in custom fields.</p> <p>To disable the property, change the default to 1. Changing the property values, and getting the property to work for the first time requires full reindex.</p>
----	--

# Advanced Jira application configuration

Jira has a number of advanced configuration options, each of which is defined as an individual property (or 'key' associated with a value). These key-value pairs are stored in one of three areas for use by Jira:

- [The Jira database](#)
- [The `jira-config.properties` file](#)
- [The `jpm.xml` file](#)

## The Jira database

The values of a small number of most commonly edited advanced configuration options are stored in the Jira database. These values can be edited from the **Advanced Settings** page of Jira's administration area. To access the values for editing, see [Configuring Jira options](#).

Once any of these properties' values are changed, they become effective immediately.

## The `jira-config.properties` file

Custom values for Jira's remaining advanced configuration options (i.e. not stored in the [Jira database](#)) are stored as individual key-value pairs in a file called `jira-config.properties` (located in the [Jira application home directory](#)). Typically, these options are of little interest to most Jira system administrators. While these key-value pairs can be edited, Jira must be restarted for any changed values to take effect.

### Example contents to demonstrate format

```
jira.projectkey.warning = testwarning
jira.projectkey.description = testdescription
```

**i** In new Jira installations, this file may not initially exist and if so, needs to be created manually. For more information about editing the `jira-config.properties` file, see [Edit the `jira-config.properties` file in Jira server](#).

## The `jpm.xml` file

Default values for all\* of Jira's available advanced configuration options are stored in a file called `jpm.xml` (located in the `<jira-application-dir>/WEB-INF/classes` subdirectory of the [Jira application installation directory](#)). These default values are only used by Jira if a property's value has not already been customized in either the [Jira database](#) (via Jira's 'Advanced Settings' page) or the `jira-config.properties` file.

**⚠ The `jpm.xml` file should not be edited** because any values that you customize in it will not be migrated automatically during subsequent Jira upgrades. To change the value of a property for an advanced configuration option in Jira, override the value of this property by redefining it in either:

- The Jira database (via Jira's 'Advanced Settings' page).
- OR**
- The `jira-config.properties` file.

\* Jira recognizes a small number of properties, which can be set in your `jira-config.properties` file *but have no definition in the `jpm.xml` file*. These properties:

- typically represent advanced configuration options that are disabled when they are not defined in your `jira-config.properties` file and
- when not specified in your `jira-config.properties` file, typically affect Jira's behavior differently to when they are specified in your `jira-config.properties` file *with no value*.

## Making changes to the `jira-config.properties` file

1. Shut down Jira (for example, by executing either the `/bin/stop-jira.sh` or `\bin\stop-jira.bat` file in your [Jira application installation directory](#), or by stopping the Jira service).
2. Open the `jira-config.properties` file (located at the root of your [Jira application home directory](#)) in a text editor.  
 This file may not exist if you are using a new Jira installation or an upgraded Jira installation where your previous Jira version(s) had never been customized. If this file does not exist, create it using a text editor.
3. Edit the appropriate properties in this file.  
 **Editing tips:**
  - To determine the default value of a property whose value you wish to redefine, search for that property in the `<jira-application-dir>/WEB-INF/classes/jpm.xml` file (of your Jira Installation Directory). The default value is defined in the `<default-value/>` sibling element of the relevant property's `<key/>` element.
  - To override a property's default value in `jpm.xml` (which is not already defined in your `jira-config.properties` file or available on the ['Advanced Settings' page](#)):
    - a. Copy the value of the relevant property's `<key/>` element from the `jpm.xml` file to the `jira-config.properties` file.
    - b. In the `jira-config.properties` file, add an '=' after that property's key, followed by your custom value.
  - To disable a custom property's value in the `jira-config.properties` file, either 'comment out' the property with a preceding '#' symbol or remove the property from the file.
4. Save your modifications to the `jira-config.properties` file.
5. Restart Jira.

### See also

[Setting properties and options on startup](#) for changes like setting available memory, disabling email, etc.

# Changing the constraints on historical time parameters in gadgets

A number of Jira gadgets show historical data from your Jira server. You can generally configure the time constraints on this data via gadget parameters, such as those parameters defining how far back should data be retrieved. For performance reasons, however, the Jira server can impose an overriding maximum limit on historical data retrieved by gadgets. These maximum limits imposed by the Jira server are defined by the following [advanced configuration options](#) in Jira and can be customized in your `jira-config.properties` file (located in the [Jira application home directory](#)).

```
jira.chart.days.previous.limit.yearly=36500
jira.chart.days.previous.limit.quarterly=22500
jira.chart.days.previous.limit.monthly=7500
jira.chart.days.previous.limit.weekly=1750
jira.chart.days.previous.limit.daily=300
jira.chart.days.previous.limit.hourly=10
```

## To update these properties:

1. Shut down your Jira server.
2. Edit your `jira-config.properties` file in your Jira home directory.
  - 📘 See [Making changes to the jira-config.properties file](#) for more information.
3. Locate these properties.
  - 📘 If any of these properties do not exist in your `jira-config.properties` file, add them to the file.
4. Update the values of these properties as desired.
5. Save your changes to the `jira-config.properties` file.
6. Restart your Jira server.

## Changing the default order for comments from ascending to descending

1. Access Jira's 'Advanced Settings' page. (See [Configuring advanced settings](#) for more information.)
2. Edit the value of the `jira.issue.actions.order` property by clicking the existing value and changing it from `asc` to `desc`
3. Click the '**Update**' button to save the new value in the Jira database.

# Limiting the number of issues returned from a search view such as an RSS feed

Jira allows you to view search results in several different formats, including Word, Excel, RSS, or XML.

A search view that returns too many issues can take a long time for Jira to complete and can use a large amount of memory. It can be a factor in [OutOfMemoryErrors](#) in Jira.

An large RSS feed of search results can be particularly problematic, because:

- the user's RSS reader will continue to make the request periodically (for example, every hour)
- since the RSS reader makes the request, not the user directly, the user is unaware that the request takes a long time or is failing

You can use the following three properties in `jira-config.properties` to limit the number of issues returned by a search view.

See [Making changes to jira-config.properties](#) for the details of how to make and apply changes to your `jira-config.properties` file.

## `jira.search.views.default.max`

The `jira.search.views.default.max` property sets a 'soft' limit on the number of issues returned. It has a default value of 1000. You can set it to 100 (for example), by specifying the following in your `jira-config.properties` file:

```
jira.search.views.default.max = 100
```

For an RSS or XML view, Jira applies the limit by appending the `tempMax` parameter to the URL of the search view. For example:

- <http://jira.atlassian.com/sr/jira.issueviews:searchrequest-xml/temp/SearchRequest.xml?type=2&pid=10240&resolution=-1&sorter/field=issuekey&sorter/order=DESC&tempMax=200>

In the above example, Jira will limit the number of issues returned to 200 (in this example).

However users can override this 'soft' default by removing the `tempMax` parameter from the URL or by increasing the value of `tempMax`.

## `jira.search.views.max.limit`

The `jira.search.views.max.limit` property sets a 'hard' limit on the number of issues returned. It has a default value of 1000. You can set this property's value to 200 (for example), by specifying the following in your `jira-config.properties` file:

```
jira.search.views.max.limit = 200
```

If a user makes an issue view request that would return more than 200 issues (in this example), Jira does not return the issues but instead returns a 403 (Forbidden) error. While the user might not be happy, it prevents Jira from consuming lots of resources and possibly running out of memory.

Make sure you set the value of `jira.search.views.max.limit` to greater than or equal to the 'soft' limit set by `jira.search.views.default.max`. Otherwise all search views that would return issues limited by the default 'soft' limit will instead return a 403 (Forbidden) error.

## `jira.search.views.max.unlimited.group`

You may have a requirement for most users to have the limit imposed on them, but a few users to be exempt from the limit. One example of this is if your Jira instance is Internet facing. You may want external (Internet) users to have the limit imposed on them, but for internal users to be able to produce unlimited search views. You can use the `jira.search.views.max.unlimited.group` property to achieve this.

The `jira.search.views.max.unlimited.group` property is disabled by default, by being either absent from your `jira-config.properties` file or present but disabled with a preceding '#'. If you enable this property in your `jira-config.properties` file, you must specify a valid group for its value or leave it empty. For example:

```
jira.search.views.max.unlimited.group = jira-administrators
```

 Users exempted from the limit via this technique will still have to add the `tempMax` parameter to the URL for an RSS or XML view, as described [above](#), in order to exceed the `jira.search.views.default.max` soft limit.

# Configuring file attachments

When file attachments are enabled, your users will be allowed to attach files and screenshots to Jira issues. This requires space on the server for storing the attachments.

File attachments are enabled by default. If you wish, you can [configure](#) the way Jira handles attachments, or disable this feature altogether. ⚠ Attachments are not stored in Jira's database and so will need to be [backed up](#) separately.

## Note:

- Your users must also have the **Create Attachments** [permissions](#) to attach files to issues.
- To allow users to attach a file *when creating a new issue*, you need to ensure that the **Attachment** field is *not hidden* within the [field configuration\(s\)](#) associated with the specific issue type(s).
- For all of the following procedures, you must be logged in as a user with the **Jira Administrators** [global permission](#).

## Configuring attachment settings

1. Choose **Administration** (⚙) > **System**.
2. Select **Advanced** > **Attachments** to open the Attachment page, which states whether attachments are on or off.

### Attachments Edit Settings ?

*i* To enable a user to attach files, ensure that the user has the **Create Attachments** permission for a particular project.

<b>Allow Attachments</b>	<b>ON</b>
<b>Attachment Path</b> The absolute or relative path to the directory under which attached files will be stored. When changing this path you will have to manually copy any existing attachments across to the new directory.	Default Directory [/data/jirastudio/jira/home/data/attachments]
<b>Attachment Size</b> The total upload size limit of attachments.	100.00 MB
<b>Enable Thumbnails</b> Enables the creation of thumbnail images of image attachments.	<b>ON</b>
<b>Enable ZIP support</b> Enables the ability for users to download all the attachments of an issue as a single ZIP file.	<b>ON</b>

3. Click the **Edit Settings** button, which opens the **Edit Attachment Settings** dialog box:

### Edit Attachment Settings

**Attachment Path**  Use Default Directory  
/data/jirastudio/jira/home/data/attachments

Disable Attachments

**Attachment Size**   
The total upload size limit in bytes.

**Enable Thumbnails**  ON  
 OFF  
Enable creation of thumbnails of image attachments.  
Attachments must be enabled to enable thumbnails.

**Enable ZIP support**  ON  
 OFF  
Enable ZIP support for attachments. Attachments must be enabled to enable ZIP support.

4. In the **Attachment Path** field, choose the **Use Default Directory** option. If you see more attachment path options than what is shown in the screenshot above, please refer to the [note below](#).  
 ⓘ As mentioned above, if you have not logged in as a user with the **Jira System Administrators global permission**, then this option will not be available to you.
5. In the **Attachment Size** field, specify the maximum attachment size. The default (per file) is 10485760 bytes (10 MB). The maximum attachment size (per file) is 2147483647 bytes (2 GB).
6. (Optional) In the **Enable Thumbnails** field, ensure that **ON** is selected if you wish to display image file attachments as thumbnails (or miniature previews) when viewing an issue. When this setting is enabled, Jira automatically creates thumbnails of the following types of image attachments:
- GIF
  - JPEG
  - PNG
- Please refer to the [info note below](#) for more information about thumbnails. If you use Linux, please refer to the [Linux note below](#).
7. (Optional) In the **Enable ZIP Support** field, ensure that **ON** is selected if you wish to view the contents of zip files attached to an issue and allow all files attached to an issue to be downloaded as a single ZIP file.
8. Click the **Update** button to update Jira's attachment settings.  
 To attach files to issues, the appropriate users, groups or project roles must first be assigned the **Create Attachments** permission for the relevant project(s).  
 To allow these users or group/project role members to delete their own attached files from issues, they must also be assigned the **Delete Own Attachments** permission for these projects too.  
 There is no need to proceed any further if:
- the permission schemes used by your project(s) already have the **Create Attachments** (and **Delete Own Attachments**) permission, or
  - your project(s) use Jira's built-in **Default Permission Scheme**.
- However, if you wish to configure these permissions, proceed with the steps in the section below.

## Configuring create/delete attachment permissions

1. Choose **Administration** (⚙️) > **Issues**.

2. Select **Permission Schemes** to open the Permission Schemes page, which displays a list of all permission schemes in your Jira system and the projects that use each scheme.
3. For each relevant permission scheme:
  - a. Click the **Permissions** link associated with the relevant permission scheme to edit that scheme's permissions.

- b. On the **Edit Permissions** page, locate **Create Attachments** within the **Attachment Permissions** section and click the **Add** link.
- c. In the user selection options on the right of the **Add New Permission** page, select the relevant (groups of) users or roles and then click the **Add** button.

- ✔ To allow these users or group/project role members to delete their own attachments, do not forget to assign them the **Delete Own Attachments** permission too.

### **i** Choosing a custom Attachment Path:

- If you upgraded Jira with an [XML backup](#) from a Jira version prior to 4.2 and used a custom directory for your attachment path, you can choose between using this custom directory (which cannot be edited) or the default directory for your attachment path location. However, once you switch to using the default directory, you can no longer choose the custom directory option.
- The default directory location is the `data/attachments` subdirectory of the [Jira home directory](#).
- To be able to change the default path, [create a symbolic link](#) to the new path.

### **i** More information about thumbnails:

- You can configure the [issue navigator](#) column layout to display the thumbnails in an `Images` column.

- All thumbnail images are stored in PNG format in the `attachments` directory, together with the original attachments. The thumbnail images are denoted by `'_thumb_'` in their file names.

### **Thumbnail image generation on Linux:**

- Your system must have X11 support. This [web page](#) details the minimum set of libraries needed to use JDK 1.4.2 under RedHat Linux 9.0.
- The following java system property must be set: `-Djava.awt.headless=true`

## Advanced configurations

You can implement the following [advanced configurations](#) to modify the way Jira handles attachments. However, these are not accessible through Jira's attachment settings. One of these advanced configurations can be modified as an 'Advanced Setting' in Jira's administration area, although the remaining two are implemented by defining properties in your `jira-config.properties` file.

### Configuring thumbnail size

By default, thumbnails are 200 pixels wide and 200 pixels high. To change the dimensions of thumbnail images:

1. Stop Jira.
2. Edit the `jira-config.properties` file in your [Jira home directory](#).
  - See [Making changes to the jira-config.properties file](#) for more information.
3. Edit the values of the following properties:
  - `jira.thumbnail.maxwidth` thumbnail width in pixels
  - `jira.thumbnail.maxheight` thumbnail height in pixels
  - If neither of these properties exist in your `jira-config.properties` file, add them to the file. For example, specify the following for a thumbnails that are 100 pixels wide:
 

```
jira.thumbnail.maxwidth = 100
```
4. Delete all existing thumbnail images within the `attachments` directory (that is, those containing `'_thumb_'` in the filename).
5. Restart Jira.

After restarting Jira, all thumbnails will be recreated automatically using the new dimensions.

### Configuring ZIP-format file accessibility

By default, Jira allows you to access common ZIP-format files, with file extensions like `.zip` and `.jar` (Java archive files). However, there are numerous other ZIP-format files to which Jira does not permit access by default. You can permit access to these files by doing the following:

1. Stop Jira.
2. Edit the `jira-config.properties` file in your [Jira home directory](#).
  - See [Making changes to the jira-config.properties file](#) for more information.
3. Remove the extensions from the `jira.attachment.do.not.expand.as.zip.extensions.list` property of the file types whose contents you wish to access in Jira.
  - If this property does not exist in your `jira-config.properties` file, add the name of this property, followed `'='`, followed by the content of the `<default-value/>` element copied from your Jira installation's `jpm.xml` file. Then, begin removing the extensions of file types whose contents you wish to access in Jira.
4. Restart Jira.

### Configuring the number of files shown in the content of ZIP-format files on issues

By default, Jira shows a maximum of 30 files in the content of ZIP-format files attached to an issue. To change this maximum value:

1. Access Jira's **Advanced Settings** page. (See [Advanced Jira configuration](#) for more information.)

2. Edit the value of the `jira.attachment.number.of.zip.entries` property by clicking the existing value and specifying the maximum number of attachments you want to show on an issue.
3. Click the **Update** button to save the new value in the Jira database.

# Configuring issue linking

## About issue linking

Issue linking allows you to create an association between issues on either the same or different Jira servers. For instance, an issue may *duplicate* another, or its resolution may *depend* on another's. New installations of Jira come with four default types of links:

- **relates to / relates to**
- **duplicates / is duplicated by**
- **blocks / is blocked by**
- **clones / is cloned by**

Issue linking also allows you to:

- Create an association between a Jira issue and a Confluence page.
- Link a Jira issue to any other web page.

You can add, edit or delete link types to suit your organization, as described below.

### Note:

- Your users must have the [Link Issues permission](#) before they can link issues.
- Issue linking must be enabled in order for your users to be able to link issues. **Issue linking is enabled by default.** If your organization does not require the ability to link issues, you can disable it globally for all users, as described [below](#).
- If you want to link Jira issues to those on a different Jira server or to Confluence pages, see [Configuring issue linking for external applications](#) (below) for details on how to set this up.
- For all of the following procedures, you must be logged in as a user with the [Jira Administrators global permission](#).

## Adding a link type

1. Choose **Administration**  **> Issues**.
2. Select **Issue Features > Issue Linking** to open the Issue Linking page.
3. In the 'Add New Link Type' form at the end of the page:
  - Enter 'Causes' in the **Name** text field.
  - Enter 'causes' in the **Outward Link Description** text field.
  - Enter 'is caused by' in the **Inward Link Description** text field.
4. Click the **Add** button.
5. This returns to the **Issue Linking** page, with a new section listing the **Causes** link type.

**Issue linking is currently ON.** ?

To deactivate issue linking, simply click below.

? For the users you wish to be able to link issues, ensure that they have the **Link Issues** permission for that particular project.

Name	Outward Description	Inward Description	Operations
<b>Blocked</b>	blocks	blocked by	<a href="#">Edit</a> · <a href="#">Delete</a>
<b>Bonfire Testing</b>	testing discovered	discovered while testing	<a href="#">Edit</a> · <a href="#">Delete</a>
<b>Cloners</b>	is cloned by	clones	<a href="#">Edit</a> · <a href="#">Delete</a>
<b>Dependent</b>	depends on	is depended on by	<a href="#">Edit</a> · <a href="#">Delete</a>
<b>Derived</b>	derived from	derives	<a href="#">Edit</a> · <a href="#">Delete</a>
<b>Design</b>	has design on	is design for	<a href="#">Edit</a> · <a href="#">Delete</a>
<b>Duplicate</b>	duplicates	is duplicated by	<a href="#">Edit</a> · <a href="#">Delete</a>
<b>Epic</b>	belongs to Epic	is the Epic for	<a href="#">Edit</a> · <a href="#">Delete</a>
<b>Relates</b>	relates to	relates to	<a href="#">Edit</a> · <a href="#">Delete</a>
<b>User Test</b>	was in test	participants were	<a href="#">Edit</a> · <a href="#">Delete</a>
<b>caused</b>	causes	caused by	<a href="#">Edit</a> · <a href="#">Delete</a>

**Add New Link Type**

Add a new link type

Name   
(eg "Duplicate")

Outward Link Description   
(eg "duplicates")

Inward Link Description   
(eg "is duplicated by")

## Editing or deleting a link type

? It is recommended that you do not edit or delete the **Clones** link type, as this is used to automatically link issues when they are cloned.

1. Choose **Administration**  **> Issues**.
2. Select **Issue Features > Issue Linking** to open the Issue Linking page.
3. Locate the link type you wish to edit or delete, and click the link type's associated **Edit/Delete** link in the **Operations** column.

## Configuring issue linking for external applications

It is possible to create links to issues on a remote Jira instance or pages on a Confluence instance (running Confluence version 4.0 or later). To do this, create *fully reciprocal application links* between your Jira instance to the remote Jira or Confluence instance. Fully reciprocal application links mean that:

1. An application link must be configured on each server to the other.
2. Each of these application links must have both **incoming and outgoing authentication** configured to each other's servers.

**To configure fully reciprocal application links between your Jira instance and a remote Jira or Confluence instance:**

1. Log in as a user with the **Jira System Administrators** [global permission](#).
2. Create an application link to your remote Jira or Confluence instance. (See [Using AppLinks to link to other applications](#) for details.) When creating the link:
  - a. During step 2 of the wizard, ensure you choose the option to create a link from the remote server back to your server.
  - b. During step 3 of the wizard, choose the **These servers fully trust each other** option. This will ensure that [incoming and outgoing authentication](#) is configured for the application link on each server to the other server.
3. If you configured a fully reciprocal application links between your Jira instance and a Confluence instance, ensure that the Confluence instance's system administrator has enabled the **Remote API (XML-RPC & SOAP)** feature, since this Confluence feature is disabled by default. See [Enabling the Remote API](#) in the Confluence documentation for details.
 

**⚠️** If you do not enable this feature, Jira will not be able to communicate with Confluence. As a result, your users:

  - a. Will see **Failed to load** messages in the Confluence Wiki page links they create on Jira issues.
  - b. Will not be able to search for Confluence pages using the **Find a Confluence page** dialog box.

**⚠️ Please Note:** You can create a one-way application link from your Jira instance to a remote Jira instance or Confluence instance. However, some loss of functionality will be experienced by your users when they create remote links. For instance, if your users create a link to a remote Jira issue, they will find that the **Create reciprocal link** check box on the **Link** dialog box will not function correctly. Hence, it is recommended that you create fully reciprocal links instead.

## Disabling issue linking

1. Choose **Administration**  **> Issues**.
2. Select **Issue Features > Issue Linking** to open the Issue Linking page.
3. A status message indicates whether issue linking is enabled. If issue linking is enabled, click the **Deactivate** button. The **Issue Linking** page reloads, stating that linking is disabled.

## Configuring the order of linked issues displayed on the 'view issue' page

Jira system administrators can define the order in which linked issues are displayed in the Issue Links section on the 'view issue' page. This is done by editing the value of the `jira.view.issue.links.sort.order` property on Jira's [Advanced settings](#) page.

Specify the fields by which to sort issues in the **Issue Links** section on the 'view issue' page by entering the appropriate 'value' for each field in a comma-separated list. This property behaves similarly to a list of values specified after the ORDER BY keyword in Jira Query Language (JQL), whereby sorting is conducted by the first and then subsequent fields specified in the list.

The `jira.view.issue.links.sort.order` property can accept the following individual field values: 'key', 'type', 'status', 'priority' and 'resolution'.

# Configuring issue cloning

Jira's issue cloning behavior can be modified by [Jira system administrators](#).

## Configuring cloned issue linking behavior

By default, when an issue is cloned, Jira will automatically create a link between the original and cloned issue using the pre-existing link type name 'Cloners'.

You can change this default behavior by editing the `jira.clone.linktype.name` property of your [jira-config.properties](#) file.

**i** If this property does not exist in your `jira-config.properties` file, add it to the file.

- If this property has a value, Jira will use the pre-existing link type whose name is the value specified for this property.
- If this property has no value, Jira will not create links between original and cloned issues.

## Configuring the cloned issue summary field prefix

By default, the 'Summary' field of a cloned issue is prefixed with the string 'CLONE - ' to indicate that the issue is a clone.

### To change this prefix or prevent the addition of prefixes on cloned issues:

1. Access Jira's Advanced Settings page. (See [Advanced Jira configuration](#) for more information.)
2. Edit the value of the `jira.clone.prefix` property by clicking the existing value and specifying a different prefix for the 'Summary' field of cloned issues.
  - i** Specifying no value prevents a prefix being added to the 'Summary' field of cloned issues.
3. Click the '**Update**' button to save the new value in the Jira database.

# Configuring the allowlist

Jira admins can allow incoming and outgoing connections and content from specified sources by adding URLs to the allowlist. Jira will display an error if content has been added that is not from an allowed source, and prompt the user to add the URL to the allowlist.

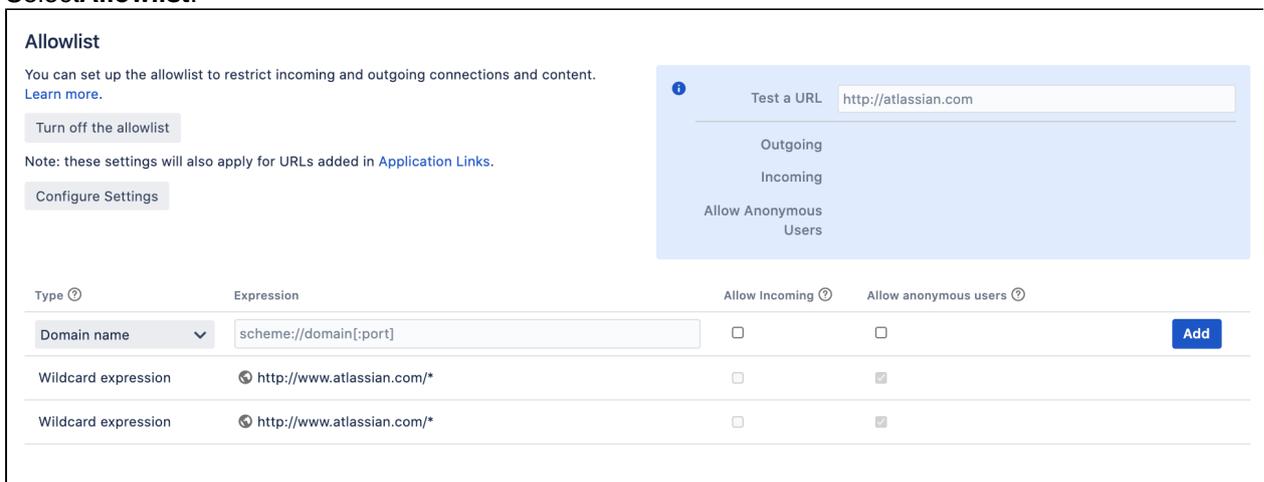
[Application Links](#) are automatically added to the allowlist. You don't need to manually add them.

 For all of the following procedures, you must be logged in as a user with the **Jira Administrators** global permission.

## Add allowed URLs to the allowlist

To add URLs to the allowlist:

1. In the upper-right corner of the screen, select **Administration**  **System**.
2. Select **Allowlist**.



3. Enter the URL or expression you want to allow.
4. Choose the **Type** of expression (see *Expression Types* below for examples).
5. Choose **Allow Incoming** if you need to allow CORS requests (see [below](#)).
6. Choose **Allow anonymous users** if you need to allow unauthenticated users.
7. Choose **Add**. Your URL or expression appears in the allowlist.

To test that your allowlisted URL is working as expected, you can enter a URL in the **Test a URL** field. Icons will indicate whether incoming or outgoing traffic is allowed for that URL.

## Expression types

When adding a URL to the allowlist, you can choose from a number of expression types.

Type	Description	Example
Domain name	Allows all URLs from the specified domain.	http://www.example.com
Exact match	Allows only the specified URL.	http://www.example.com/thispage
Wildcard Expression	Allows all matching URLs. Use the wildcard * character to replace one or more characters.	http://*example.com
Regular Expression	Allows all URLs matching the regular expression.	http(s)?://www\.example\.com

## Allow incoming

**Allow Incoming** enables **CORS** requests from the specified origin. The URL must match the format `scheme://host[:port]`, with no trailing slashes (`:port` is optional). So `http://example.com/` would not allow CORS requests from the domain `example.com`.

## Allow anonymous users

You can use the **Allow anonymous users** option to allow outbound requests on behalf of unauthenticated users.

This isn't recommended for URLs that may contain private data, such as URLs from application links. If you do need to provide anonymous access, consider using an exact URL or wildcard based rule to limit access to just the required resources.

## Change default settings for new application links

When you create an application link, the URL is automatically added to the Jira allowlist. By default, outbound requests from these URLs are only allowed for authenticated users.

To change the default behavior for all application links:

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. Select **Allowlist**.
3. Select **Configure Settings**.
4. Select either:
  - **Allow all users** to allow outbound requests for all users, including anonymous users
  - **Allow authenticated users** to deny outbound requests for anonymous users
  - **Restrict by default** to deny outbound requests for all users (application link won't be added to the allowlist)
5. **Save** your changes.

All application links, including new application links added to the allowlist, will use this setting.

## Disable the allowlist

To disable the allowlist:

The allowlist is enabled by default. You can choose to disable it, but this will allow all URLs, including malicious content, and is not recommended.

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. Select **Security > Allowlist**.
3. Click the **Turn off allowlist** button.
4. Choose **Confirm**.

All URLs will now be allowed. Unless your instance is running in an environment without internet access, we do not recommend disabling the allowlist.

# Configuring sub-tasks

*Sub-task issues* are generally used to split up a parent issue into a number of tasks which can be assigned and tracked separately.

Sub-tasks have all the same fields as standard issues, although note that their 'issue type' must be one of the *sub-task issue types* (see below), rather than one of the standard issue types.

If sub-tasks are enabled and you have defined at least one sub-task issue type, your users will be able to:

- create sub-tasks
- convert issues to sub-tasks (and vice versa)

## On this page:

- [Disabling sub-tasks](#)
- [Enabling sub-tasks](#)
- [Defining sub-task issue types](#)
- [Blocking issue workflows by sub-task status](#)
- [Configuring sub-task fields displayed on parent issues](#)

## Disabling sub-tasks

Sub-tasks are enabled by default. However, this feature can be disabled from the Sub-Tasks administration page.

 *Sub-tasks will be disabled by default if your Jira installation was upgraded from a version prior to 4.2 that had sub-tasks disabled.*

1. Log in as a user with the [Jira Administrators global permission](#).
2. Choose **Administration**  **> Issues**. Select **Issue Types > Sub-Tasks** to open the Sub-Tasks page.
3. Click the **'Disable' Sub-Tasks** link. The page reloads and informs you that sub-tasks are now disabled.

 **Please note:** Sub-tasks cannot be disabled if one or more sub-tasks exists in the system. You must remove any existing sub-tasks (or convert them to standard issues) before you can disable this feature.

## Enabling sub-tasks

Sub-tasks can be enabled from the Sub-Tasks administration screen.

1. Log in as a user with the [Jira Administrators global permission](#).
2. Choose **Administration**  **> Issues**. Select **Issue Types > Sub-Tasks** to open the Sub-Tasks page.
3. Click the **'Enable' Sub-Tasks** link. The page will reload and inform you that the sub-tasks are now enabled.
  -  A default [sub-task issue type](#) is automatically available for use. You can edit it by clicking its **Edit** link in the Operations column.

## Defining sub-task issue types

Sub-tasks must be assigned one of the *sub-task issue types*, which are different to standard issue types. Please note that at least one sub-task issue type must be defined in Jira for users to be able to create sub-tasks.

Sub-task issue types can be customized on the Sub-Tasks administration page (described above). The Sub-Tasks administration page also allows you to create, edit (i.e. the name, description or icon), and translate your sub-task issue types.

### Creating a sub-task issue type

1. Log in as a user with the [Jira Administrators global permission](#).
2. Choose **Administration**  **> Issues**. Select **Issue Types > Sub-Tasks** to open the Sub-Tasks page.

## Sub-Tasks

[+ Add New Sub-Task Issue Type](#)

Sub-Tasks are currently turned **ON**. You can manage your sub-tasks as part of standard issue types [here](#).

- [Disable Sub-Tasks](#)
- [Translate Sub-Tasks](#)
- [Manage Sub-Tasks](#)

Name	Description	Icon	Operation
Sub-task	The sub-task of the issue		<a href="#">Edit</a>

3. Click **Add New Sub-task Issue Type** button to open the Add New Sub-task Issue Type dialog box.
4. Complete the following:

- **Name** enter a short phrase that best describes your new sub-task issue type.
- **Description** enter a sentence or two to describe when this sub-task issue type should be used.
- **Icon URL** supply the path of an image from an accessible URL or an image that has been placed somewhere inside `<jira-application-dir>/images/icons` of your [Jira application installation directory](#).

### Editing a sub-task issue type

1. Log in as a user with the [Jira Administrators global permission](#).
2. Choose **Administration**  **> Issues**. Select **Issue Types > Sub-Tasks** to open the Sub-Tasks page.
3. Click the **Edit** link (in the Operations column) for the sub-task issue type that you wish to edit.
4. Edit the **Name**, **Description**, and/or **Icon**, as described above in [Creating a sub-task issue type](#).

### Deleting a sub-task issue type

You can only delete sub-task issue types through the Manage Issue Types page. For details, see [Deleting an issue type](#).

### Blocking issue workflows by sub-task status

It is possible to restrict the progression of an issue through workflow depending on the status of the issue's sub-tasks. For example, you might need to restrict an issue from being resolved until all of its sub-tasks are resolved. To achieve this, you would create a [custom workflow](#) and use the *Sub-task Blocking Condition* on the workflow transitions that are to be restricted by the sub-tasks' status.

### Configuring sub-task fields displayed on parent issues

Jira system administrators can define which fields of sub-tasks are displayed in the Sub-tasks section on the 'view issue' page of a parent issue (which contains one or more sub-tasks). This is done by editing the value of the `jira.table.cols.subtasks` property on Jira's [Configuring advanced settings](#) page.

Specify which fields you want to show in the **Sub-tasks** section of a parent issue's 'view issue' page by entering the appropriate 'value' for each field in a comma-separated list. The `jira.table.cols.subtasks` property can accept the values indicated in right-hand column of the `IssueFieldConstants` table on the [Constant Field Values](#) page (of Jira's API documentation).

#### Please note:

- The order of each value in this list determines the order of their representative fields in the Sub-tasks section of a parent issue's 'view issue' page.
- The `summary` field is a mandatory value which assumes first position in this property's value.

 **Do more with Jira**

Extend what you can do with sub-tasks using these apps from the [Atlassian Marketplace](#):

- [Create on Transition for Jira](#): Automatically create multiple issues and sub-tasks when a select issue transitions through a workflow
- [Quick Subtasks for Jira](#): Create multiple sub-tasks using a simple text syntax

# Managing filters

A filter is a saved issue search. You can search for issues using different criteria in basic or advanced search, and then save your search criteria as a filter, becoming the filters owner. The owner can decide what to do with their filter either make it private for personal use or share it with different entities, such as users, projects, or groups. If the filter is private, only the owner and Jira admin can view and modify it.

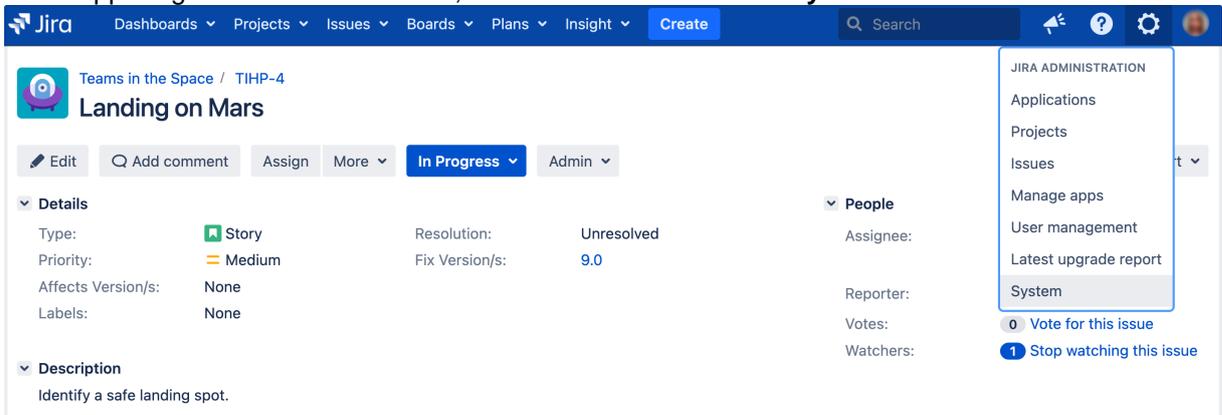
**i** For all of the following procedures, you must be logged in as a user with the **Jira system administrator** permissions. For details, see [Permissions overview](#).

You don't need admin permissions to create and configure your own Jira filters. Check out [Saving your search as a filter](#) for detailed instructions on how to manage filters as a user.

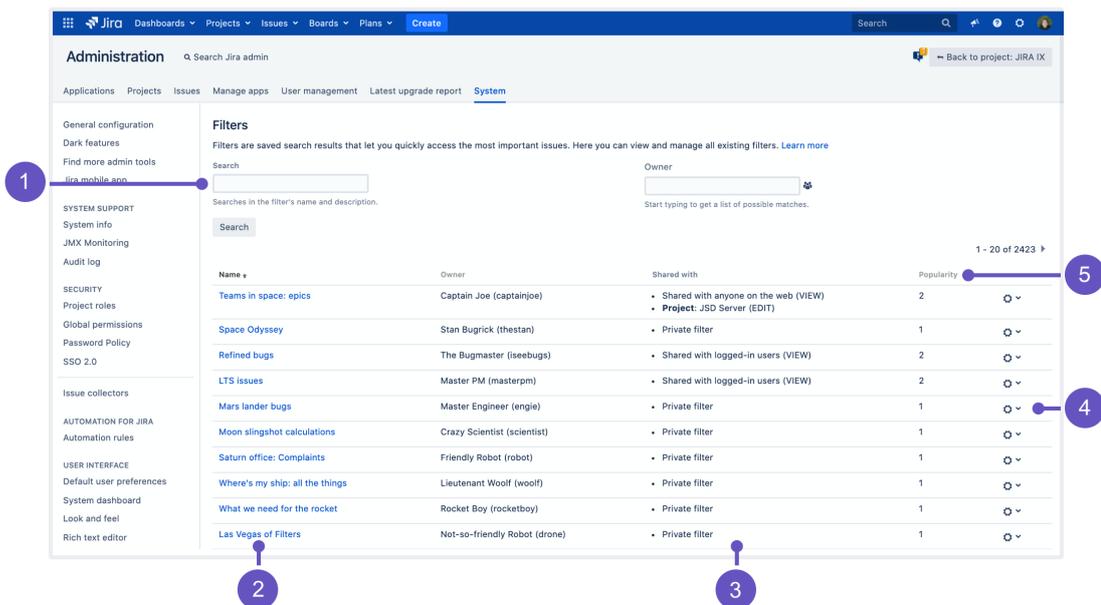
## Viewing filters

You can view and manage all existing filters, both private and shared, from the Jira Administration menu.

1. In the upper-right corner of the screen, select **Administration** > **System**.



2. Under the **Shared items** (the left-side panel), select **Filters**.



On the **Filters** page, you can view and configure both private and shared filters:

1. **Search/Owner:** Search existing filters by name or description, or enter the username to see all filters belonging to a user.
2. **Name:** Select the filter name to open it. From here, you can edit the filters search criteria, add or change existing shares, and manage the subscriptions.

3. **Shared with:** Check the users or other entities the filter is shared with. For more information, see [Understanding shares](#).
4. **Actions:** Change the filters ownership or delete the filter. See [Deleting filters or changing their owners](#) for details.
5. **Popularity:** Check the number of users who have selected this filter as their favorite.

If you experience problems when trying to view or use a filter, check out the following article: [Field or Filter does not exist or you do not have permission to view it](#).

## Editing filters

You can change the filter's search criteria, owner, permissions, or subscriptions.

1. Select the filter name to view the search page.
2. Select **Details** to view quick info about the filter. From there, you can:
  - Edit permissions add or change viewers and editors.
  - Edit subscription add or change subscriptions.

The screenshot shows the Jira interface with a 'Details' modal open for a filter. The modal contains the following information:

- Owned by:** Friendly Robot
- Who can view this filter:** everybody
- Who can edit this filter:**
  - Projects: JIRA IX, Teams in Space
  - Users: Master Engineer, Crazy Scientist
- Edit permissions:** (link)
- Subscriptions:** This filter has no subscriptions. (link) [New subscription](#)

The background shows the 'Space Mission' filter page with a search bar and a list of issues.

## Deleting filters or changing their owners

You can change the filters ownership or delete it altogether. This is useful if:

- the filters owner is no longer with your organization.
- the filter returns too many issues, which can affect performance.

✔ Before changing the owner or deleting a filter, its good to inform the filters current owner of your intentions.

To delete a filter:

1. Select **Actions** (⚙️) menu next to the filters name.
2. Select **Change owner** or **Delete**.

ⓘ If your filter is used on Jira boards or has subscribed users, you wont be able to remove it until you delete these boards and subscriptions. Be careful when deleting the filter as it might affect other peoples work.

## Understanding the shares

In one of the columns, **Shared with**, you can check who can view or edit a filter. Here are the types of entities that a filter can be shared with, just so you know what you're managing.

<b>Shared with</b>	<b>Description</b>
<b>Private</b>	The filter is private and can only be viewed and edited by the owner and Jira admins.
<b>User /Group /Project</b>	The filter is shared with a user, group, or project. Depending on the permissions granted, people the filter is shared with can view or edit the filter.
<b>Any logged-in user</b>	The filter is shared with all users on your Jira instance.
<b>Anyone on the web</b>	<p>If a filter is <b>Shared with anyone on the web</b>, this allows public access to certain information on your site. If you want to prevent people who aren't logged in from viewing this information and limit access to your logged in users, either:</p> <ul style="list-style-type: none"> <li>• change the filter's permissions.</li> <li>• remove public access to these filters. See <a href="#">Control anonymous user access</a>.</li> </ul>

# Managing dashboards

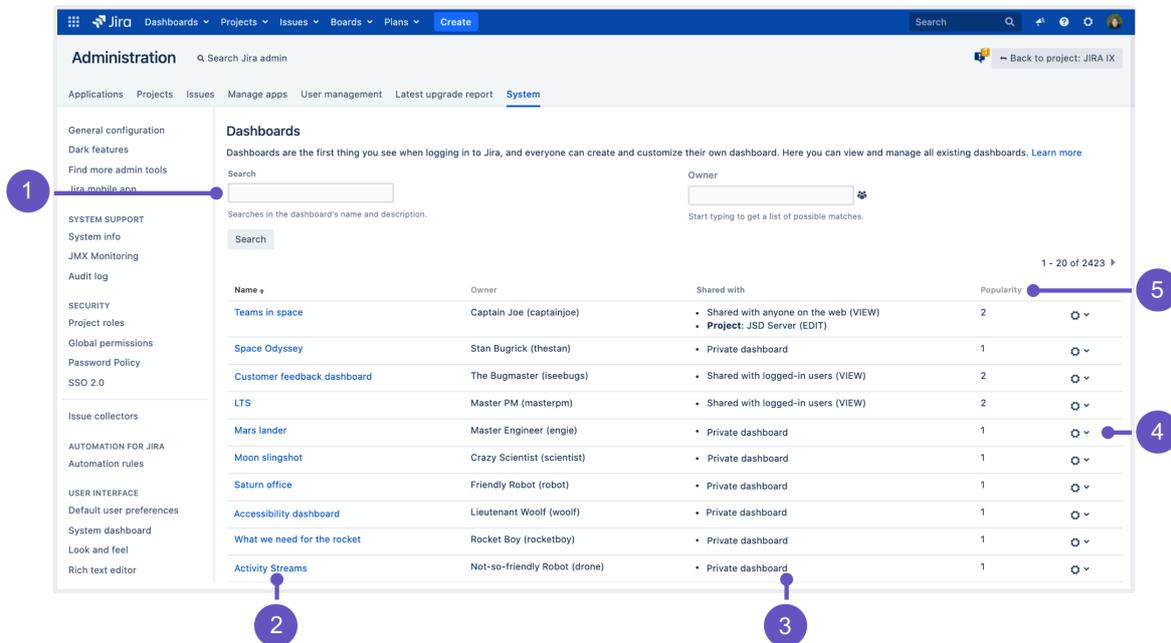
Dashboards are the first thing you see when logging in to Jira. Every user can create and customize their own dashboard with their favorite gadgets. The owner of the dashboard can decide what to do with it either make it private for personal use or share it with different entities, such as users, projects, or groups. If the dashboard is private, only the owner and Jira admin can view and modify it.

 You need to be a Jira administrator to complete all of the following tasks.

## Viewing dashboards

You can view and manage all existing dashboards, whether they're shared or private, from the administration area.

1. Go to **Administration > System**.
2. Select **Dashboards** in the sidebar.

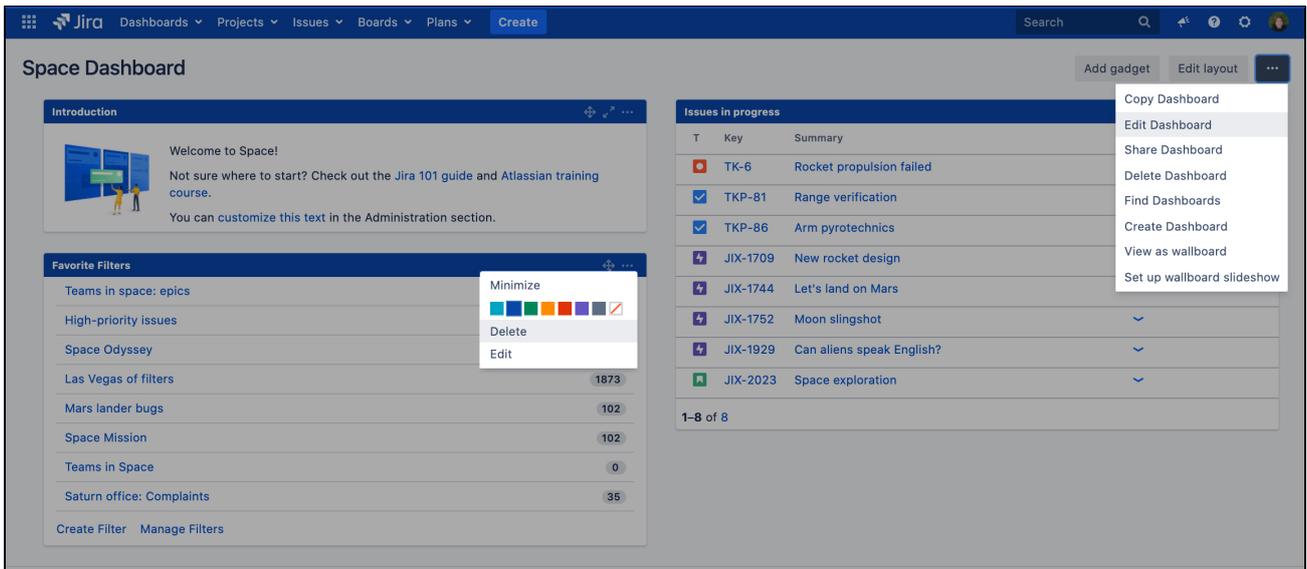


1. **Search/Owner:** Search existing dashboards by name or description, or enter the username to see all dashboards belonging to a user.
2. **Name:** Select the dashboard name to open it. You'll be able to edit the dashboard's details and modify the gadgets that appear on it.
3. **Shared with:** Check the users or other entities the dashboard is shared with. For more info on what they mean, see [Understanding shares](#).
4. **Actions:** Change the dashboard's ownership, or delete it.
5. **Popularity:** Check the number of users who have selected this dashboard as their favorite.

## Editing dashboards

You can edit the gadgets displayed on the dashboard or change the dashboard's permissions and owners.

1. Select the dashboard name. The dashboard will open.
2. From there, you can:
  - a. Edit the settings of every gadget that appears on the dashboard.
  - b. Edit the dashboard itself, adding or changing the shares.



### Deleting dashboards and changing their owners

You can change the dashboards ownership or delete it altogether. This is useful if:

- The dashboards owner is no longer with your organization
- The dashboard includes gadgets that affect the performance of your instance

✔ Before changing the owners or deleting a dashboard, its good to inform the dashboard's current owner of your intentions.

1. Select **Actions (icon)** next to the dashboard's name.
2. Select **Change owner** or **Delete**.

### Understanding the shares

Here are the available options for sharing a dashboard.

Shared with	Description
<b>Private</b>	The dashboard is private and can only be viewed and edited by the owner and Jira admins.
<b>User / Group / Project</b>	The dashboard is shared with a user, group, or a project. Depending on the permissions granted, people the dashboard is shared with can view or edit the dashboard.
<b>Any logged-in user</b>	The dashboard is shared with all users on your Jira instance.
<b>Anyone on the web</b>	If a dashboard is <b>Shared with anyone on the web</b> , this allows public access to certain information on your site. If you want to prevent people who arent logged in from viewing this information and limit access to your logged in users, either: <ul style="list-style-type: none"> <li>• change the dashboards permissions</li> <li>• remove public access to these dashboards. See <a href="#">Control anonymous user access</a>.</li> </ul>

# Enabling logout confirmation

Administrators can configure Jira to prompt users with a confirmation before logging them out.

**Note:** For all of the following procedures, you must be logged in as a user with the **Jira Administrators** [global permission](#).

By default, Jira will not prompt users to confirm logging out. To change this:

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. Select **General configuration** to open the Administration page.
3. Locate the 'Options' section.

By default, Jira will not prompt users to confirm logging out. To change this, click the **Edit Settings** button at the top of the page, and then enable or disable logout confirmation.

The **Never** and **Always** settings are self-explanatory. When set to **Cookie**, your Jira users will only be prompted if they have logged in using a cookie (i.e. by selecting the '**Remember my login on this computer**' checkbox before they click the '**Log In**' button).

# Rich text editing

**i** For all of the following procedures, you must be logged in as a user with the **Jira System administrator** permissions. For details, see [Permissions overview](#).

The rich text editor lets your users choose between two options:

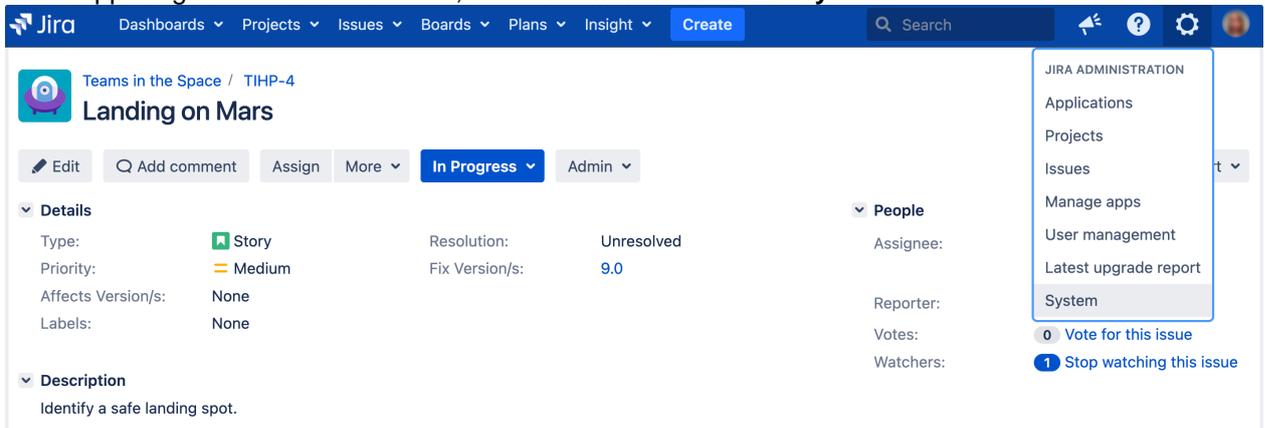
- **Text** mode, which supports a modified version of wiki markdown. [Learn more about Jira markdown formatting syntax](#).
- **Visual** mode, which is a What You See Is What You Get (WYSIWYG) editor.

The rich text editor is available on description fields, comment fields, and all Text field (multi-line) custom fields that use the [wiki renderer](#).

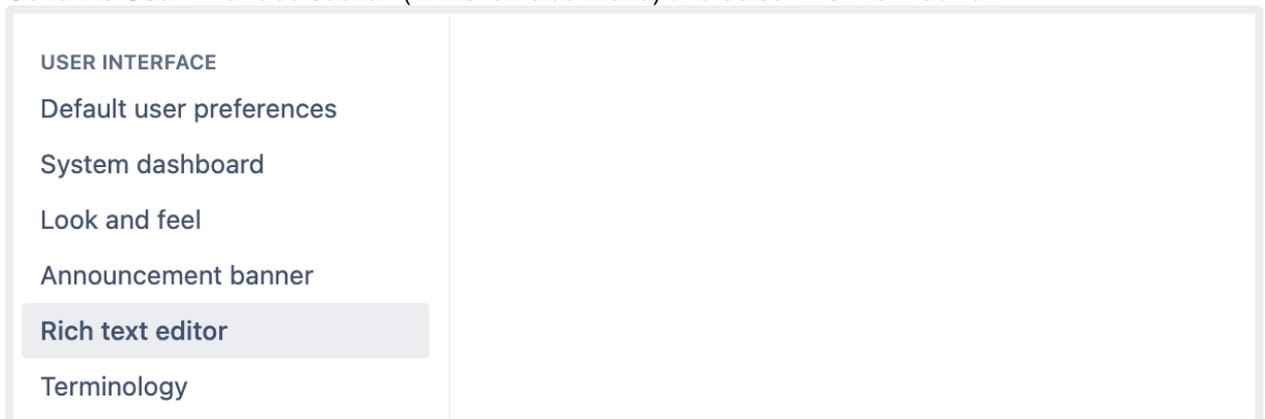
## Enable or disable the rich text editor

Rich text editing is enabled by default, but you can disable it as follows:

1. In the upper-right corner of the screen, select **Administration**  **System**.



2. Go to the **User Interface** section (in the left-side menu) and select **Rich text editor**.



### 3. Select the **Enable rich text editing for users** toggle to enable or disable the editor.

Manage apps User management Latest upgrade report **System**

#### Rich text editor

WYSIWYG (what you see is what you get) is becoming the standard editing experience, and the rich text editor allows you to give your users the option of using wiki markdown in Text mode, or a WYSIWYG experience in Visual mode. Let your users choose how they prefer to create and edit their content.

Enabling the rich text editor allows users to choose their editing experience, wiki markdown or rich text. The rich text editor is applicable to description fields, comment fields and Text field (multi line) custom fields that are configured to use the [wiki renderer](#).

Enable rich text editor for users

Please make sure that fields with rich text editor are using Wiki style renderer in their configuration

[Learn more about the rich text editor.](#)

Currently, the editor doesn't support:

- nested tables
- pasting rich text (plain text is fine) that contains complex formatting

Third-party macros provided by apps that aren't compatible with Jira 8.20 are displayed in legacy mode:

Comment

Style | B | I | U | A | ↻A | | U | | ☰ | ☰

{custommacro:param=Value}

Custom macro content in wiki markup \*bold\*

Visual | Text

The Macro header is not editable in Visual mode, and content within the macro is presented in text mode (wiki markup).

You can check the status of your apps on the **Rich text editor** configuration page:

Add-ons User management **System**

### Rich text editor



---

**Warning: conflicting plugins detected**

Third-party plugins that extend wiki markdown rendering are installed on this instance. Disable or uninstall these plugins before enabling the rich text editor:

- Legacy Plugin 1
- Legacy Plugin 2

---

Enabling the rich text editor allows users to choose their editing experience, wiki or rich text editor.

Enable rich text editor for users

Please make sure that fields with rich text editor are properly configured.

[Learn more about the rich text editor.](#)

# Configuring terminology

To help you follow Agile best practices, Jira Data Center provides Flexible Terminology. We've enabled admins to change generic Jira Data Center terms. You can keep consistent naming of sprints and epics between Jira and the Agile at Scale Frameworks including SAFe and LeSS. Terminology changes apply to any variant of English.

With this functionality, terms will also be replaced in custom field names and issue types, and everywhere else in Jira, for example in reports, issues, and search results. Jira will keep capitalization of original terms whenever it's possible.

We've added a new page to help admins define new terms. Regardless of the framework you follow, or the terminology you choose, you can quickly implement changes across your instance.

## Defining terms

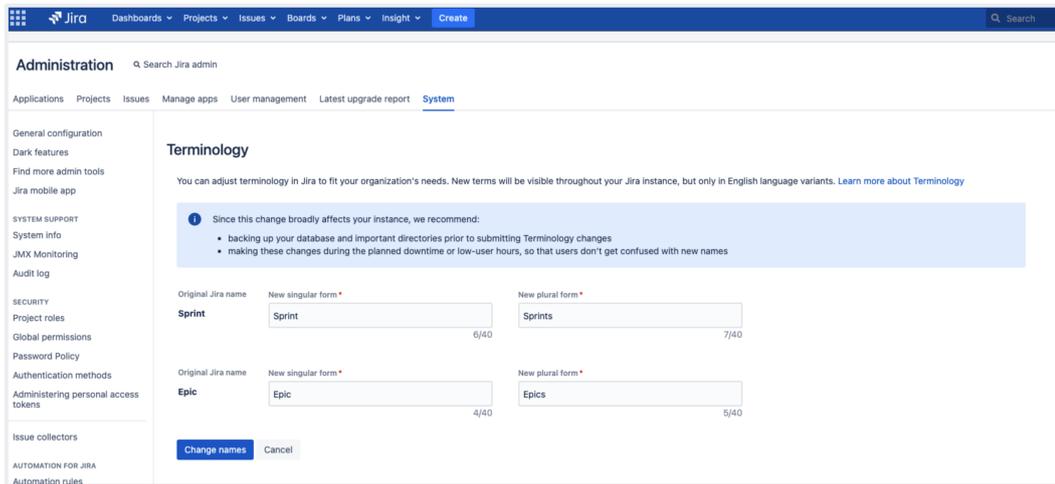
### Before you begin

Terminology change broadly affects your instance. We recommend:

- backing up your database and important directories prior to submitting terminology changes
- making these changes during planned downtime or low-user hours, so that users don't get confused by the new names

To view and manage your terms:

1. Go to **Administration > System**.
2. Click **Terminology**.
3. Define new singular and plural forms of your terms.
4. Click **Change names**.



The screenshot shows the Jira Administration interface. The top navigation bar includes 'Dashboards', 'Projects', 'Issues', 'Boards', 'Plans', 'Insight', and 'Create'. The main content area is titled 'Administration' and has a search bar. Below the navigation, there are tabs for 'Applications', 'Projects', 'Issues', 'Manage apps', 'User management', 'Latest upgrade report', and 'System'. The 'System' tab is active, and the 'Terminology' section is selected. A blue information box contains the following text: 'Since this change broadly affects your instance, we recommend: • backing up your database and important directories prior to submitting Terminology changes • making these changes during the planned downtime or low-user hours, so that users don't get confused with new names'. Below this, there are two rows of input fields. The first row is for 'Sprint', with 'Original Jira name' set to 'Sprint', 'New singular form' set to 'Sprint' (6/40 characters), and 'New plural form' set to 'Sprints' (7/40 characters). The second row is for 'Epic', with 'Original Jira name' set to 'Epic', 'New singular form' set to 'Epic' (4/40 characters), and 'New plural form' set to 'Epics' (5/40 characters). At the bottom of the form, there are 'Change names' and 'Cancel' buttons.

You can define singular and plural forms of your new terms according to the following rules:

- You can't swap names of epics and sprints.
- There is a 40-character limit.
- Terms must be unique (e.g., epics and sprints can't both be called Potatoes) and can contain only letters, numbers, and spaces.
- All fields must have a value (at least one char).

 Changing names may override translated terms in UK and US English language variants.

## Adjusting terms across Atlassian products

Admins have control over Jira terms and can define them globally without third-party apps. But, there are some things you need to consider for other products:

- **Advanced Roadmaps:** Whenever you change the term **sprint** in Jira, it will also be changed in Advanced Roadmaps. But, **epic** won't refresh automatically. You need to update the level name separately. To do this, go to **Plans > Administration > Hierarchy levels**.
- **Jira Align:** Your Jira connector doesn't sync platform terminology, so you need to update terms in Jira Align separately. To reflect the **sprint** and **epic** term update in Jira Align, go to **Admin > Platform Technology**.

## Limitations

1. Terminology changes apply only to English language variants. For other languages, Jira displays original names.
2. Flexible Terminology uses translations of Custom Field and Issue Type to localize their UI names. As a result, all REST APIs will return localized names when querying the API or saving the issue (for example, new term). Also, new change items will be persisted with translated fields name so querying for a field's history should take into account all current and historical names of the field. If your app or script has a hardcoded reference to an issue type named **epic** or custom fields with the following names:
  - Epic Name
  - Epic Color
  - Epic Status
  - Epic Link
  - Sprint

These need to be adapted to use IDs (recommended) or new names. You can do that by replacing **epic** or **sprint** with new terms, as returned by `rest/api/2/terminology/entries`.

If your app and scripts are already using custom translations for Custom Fields or Issue Types, you don't need to make any changes.

# Server optimization

This section of the documentation includes information on how to optimize your Jira installation, such as performance testing and using the configuration tool. While not included in this section of the documentation, you may also be interested in our [Tuning database connections](#) page in the [Installation](#) section.

- [Configuring secure administrator sessions](#)
- [Improving instance stability with rate limiting](#)
- [Jira application cookies](#)
- [Preventing security attacks](#)
- [Using the Jira application configuration tool](#)
- [Running Jira applications as a Windows service](#)
- [Tuning garbage collection \(GC\)](#)

If you're looking for very specific information regarding your setup, and can't find it in the documentation, you may also wish to check out the [Jira Knowledge Base](#), and [Atlassian Answers](#).

# Configuring secure administrator sessions

Jira protects access to its administrative functions by requiring a secure administration session in order to use the Jira administration screens. (This is also known as websudo.) When a Jira administrator (who is logged into Jira) attempts to access an administration function, they are prompted to log in again. This logs the administrator into a temporary secure session that grants access to the Jira administration screens.

The temporary secure session has a rolling timeout (defaulted to 10 minutes). If there is no activity by the administrator in the Jira administration screens for a period of time that exceeds the timeout, then the administrator will be logged out of the secure administrator session (note that they will remain logged into Jira). If the administrator does click an administration function, the timeout will reset.

Note that Project Administration functions (as defined by the ['Project Administrator' permission](#)) do not require a secure administration session.

## Manually ending a secure administrator session

An administrator can choose to manually end their secure session by clicking the **'drop access'** link in the banner displayed at the top of their screen.

## Disabling secure administrator sessions

Secure administrator sessions (i.e. password confirmation before accessing administration functions) are enabled by default. If this causes issues for your Jira instance (e.g. if you are using a custom authentication mechanism), you can disable this feature by specifying the following line in your [jira-config.properties](#) file:

```
jira.websudo.is.disabled = true
```

 You will need to restart your Jira server for this setting to take effect.

## Changing the timeout

To change the number of minutes of inactivity after which a secure administrator session will time out, specify the `jira.websudo.timeout` property (in your [jira-config.properties](#) file) whose value is the number of minutes of inactivity required before a secure administration session times out.

For example, the following line in your `jira-config.properties` file will end a secure administration session in 10 minutes:

```
jira.websudo.timeout = 10
```

 You will need to restart your Jira server for this setting to take effect.

## Developer notes

If you have written a plugin that has webwork actions in the Jira Administration section, those actions should have the `@WebSudoRequired` annotation added to the class (not the method or the package, unlike Confluence).

Please also see [How do I develop against Jira with Secure Administrator Sessions?](#) and [Adding WebSudo Support to your Plugin](#).

### On this page:

- [Manually ending a secure administrator session](#)
- [Disabling secure administrator sessions](#)
- [Changing the timeout](#)
- [Developer notes](#)

# Improving instance stability with rate limiting

When automated integrations or scripts send requests to Jira in huge bursts, it can affect Jira's stability, leading to drops in performance or even downtime. With rate limiting, you can control how many external REST API requests automations and users can make and how often they can make them, making sure that your Jira instance remains stable.

 Rate limiting is available for **Jira Software Data Center** and **Jira Service Management Data Center**.

## Skip to

- [How rate limiting works](#)
- [How to turn on rate limiting](#)
- [Limiting requests what it's all about](#)
- [Adding exemptions](#)
- [Identifying users who have been rate limited](#)
- [Getting rate limited user's perspective](#)
- [Allowlisting URLs and external applications](#)
- [Adjusting your code to rate limiting](#)

## How rate limiting works

Here are some details about how rate limiting works in Jira.

Rate limiting targets only external REST API requests, which means that requests made within Jira are not limited in any way. When users move around the Jira user interface, viewing projects, transitioning issues, and completing other actions, they won't be affected by rate limiting, as we're seeing this as a regular user experience that shouldn't be limited.

Let's use an example to better illustrate this:

- When a user views an issue in Jira, a number of requests are sent in the background these requests ask Jira for comments, assignees, attachments, etc. Since this traffic is internal to Jira, it won't be limited.
- When the same user opens up the terminal on their laptop and sends a request (like the one below) to get information about an issue, it will be rate limited because it's made outside of Jira.

```
wget https://jira.com/rest/api/2/issue/JRASERVER-70350
```

### Authentication mechanisms

To give you more details on how we recognize which requests should be limited, we're targeting external HTTP requests with these authentication mechanisms:

- Basic auth
- OAuth
- JSESSIONID cookie

Out of the many available techniques for enforcing rate limits, we've chosen to use **token bucket**, which gives users a balance of tokens that can be exchanged for requests. Here's a summary of how it works:

- Users are given tokens that are exchanged for requests. One token equals one request.
- Users get new tokens at a constant rate so they can keep making new requests. This is their *Requests allowed*, and can be, for example, 10 every 1 minute.
- Tokens are added to a user's personal bucket until it's full. This is their *Max requests* and allows them to adjust the usage of tokens to their own frequency, for example 20 every 2 minutes instead of 10 every 1 minute, as specified in their usual rate.

- When a user tries to send more requests than the number of tokens they have, only requests that can draw tokens from the bucket will be successful. The remaining ones will end in a 429 error message (too many requests). The user can retry those requests once they get new tokens.

Jira tastes best when used with our other products like Confluence, Bitbucket, or Bamboo. Technically, products like these are external to Jira, so they should be limited. In this case, however, were treating them as belonging to the same user experience and dont want to enforce any limits for requests coming from or to these products.

The way it is now:

- **Server:** Not limited in any way.
- **Cloud:** Theres a known issue that applies rate limits to requests coming from/to cloud products. Were working hard to disable rate limits for cloud products and should make that happen soon. For now, you can use a workaround. For more info, see [Removing rate limits for Atlassian cloud products](#).

The general assumption is that Marketplace apps are installed on a Jira instance, make internal requests from within Jira, and shouldnt be limited. But, as always, **it depends on how an app works**.

- **Internal:** If an app in fact works internally, enhancing the user experience, it wont be limited. An example of such app would be a special banner thats displayed on a Scrum board. Lets say this banner checks all issues that were done and shows this sprints winner a user whos completed the most issues in this sprint. Traffic like that would be internal, not limited.
- **External:** Apps whose requests are external to Jira are limited. Lets say we have an app that displays a wallboard on TV. It asks Jira for details about boards, issues, assignees, etc. and then reshuffles and displays them in its own way as the earlier mentioned wallboard. An app like that sends external requests and behaves just like a user sending requests over a terminal.

It really depends on the app, but were assuming most of them shouldnt be limited.

Rate limiting is available for Data Center, so you most likely have a cluster of nodes behind a load balancer. You should know that each of your users will have a separate limit on each node (rate limits are applied per node, not per cluster).

In other words, if they have used their *Requests allowed* on one node and were rate limited, they could theoretically send requests again if they started a new session on a different node. Switching between the nodes isnt something users can do, but keep in mind that this can happen.

Whatever limit youve chosen (e.g. 100 requests every 1 hour), the same limit will apply to each node, you dont have to set it separately. This means that each users ability to send requests will still be limited, and Jira will remain stable regardless of which node their requests are routed to.

Setting the right limit depends on many factors, so we cant give you a simple answer. We have some suggestions, though.

### Finding the right limit

The first step is to understand the size of traffic that your instance receives. You can do this by parsing the access log and finding a user than made the most REST requests over a day. Since UI traffic is not rate limited, this number will be higher than what you need as your rate limit. Now, thats a base number you need to modify it further based on the following questions:

1. Can you afford to interrupt your users work? If your users integrations are mission-critical, consider upgrading your hardware instead. The more critical the integrations, the higher the limit should be consider multiplying the number you found by two or three.
2. Is your instance already experiencing problems due to the amount of REST traffic? If yes, then choose a limit thats close to the base number you found on a day when the instance didnt struggle. And if youre not experiencing significant problems, consider adding an extra 50% to the base number this shouldnt interrupt your users and you still keep some capacity.

In general, the limit you choose should keep your instance safe, not control individual users. Rate limiting is more about protecting Jira from integrations and scripts going haywire, rather than stopping users from getting their work done.

## How to turn on rate limiting

**i** You need to be a Jira System Administrator to turn on rate limiting.

To turn on rate limiting:

1. In Jira, go to **Administration > System > Rate limiting**.
2. Change the status to **Enabled**.
3. Select one of the options: **Allow unlimited requests**, **Block all requests**, or **Limit requests**. The first and second are all about allowlisting and blocklisting. For the last option, you'll need to enter actual limits. You can read more about them below.
4. Click **Save**.
5. Make sure to add exemptions for users who really need those extra requests, especially if you've chosen allowlisting or blocklisting. See Adding exemptions.

## Limiting request what its all about

As much as allowlisting and blocklisting shouldn't require additional explanation, you'll probably be using the **Limit requests** option quite often, either as a global setting or in exemptions.

Lets have a closer look at this option and how it works:

1. **Requests allowed:** Every user is allowed a certain amount of requests in a chosen time interval. It can be 10 requests every second, 100 requests every hour, or any other configuration you choose.
2. **Max requests (advanced):** Allowed requests, if not sent frequently, can be accumulated up to a set maximum per user. This option allows users to make requests at a different frequency than their usual rate (for example, 20 every 2 minutes instead of 10 every 1 minute, as specified in their rate), or accumulate more requests over time and send them in a single burst, if that's what they need. Too advanced? Just make it equal to *Requests allowed*, and forget about this field nothing more will be accumulated.

## Examples

### Requests allowed: 10/hour | Max requests: 100

- One of the developers is sending requests on a regular basis, 10 per hour, throughout the day. If they try sending 20 requests in a single burst, only 10 of them will be successful. They could retry the remaining 10 in the next hour when they're allowed new requests.

- Another developer hasn't sent any requests for the past 10 hours, so their allowed requests kept accumulating until they reached 100, which is the max requests they can have. They can now send a burst of 100 requests and all of them will be successful. Once they used up all available requests, they have to wait for another hour, and they'll only get the allowed 10 requests.
- If this same developer sent only 50 out of their 100 requests, they could send another 50 right away, or start accumulating again in the next hour.

### Requests allowed: 1/second | Max requests: 60

- A developer can choose to send 1 request every second or 60 requests every minute (at any frequency).
- Since they can use the available 60 requests at any frequency, they can also send all of them at once or in very short intervals. In such a case, they would be exceeding their usual rate of 1 request per second.

## Finding the right limit

Setting the right limit depends on many factors, so we can't give you a simple answer. We have some suggestions, though.

### Finding the right limit

The first step is to understand the size of traffic that your instance receives. You can do this by parsing the access log and finding a user that made the most REST requests over a day. Since UI traffic is not rate limited, this number will be higher than what you need as your rate limit. Now, that's a base number you need to modify it further based on the following questions:

1. Can you afford to interrupt your users' work? If your users' integrations are mission-critical, consider upgrading your hardware instead. The more critical the integrations, the higher the limit should be. Consider multiplying the number you found by two or three.
2. Is your instance already experiencing problems due to the amount of REST traffic? If yes, then choose a limit that's close to the base number you found on a day when the instance didn't struggle. And if you're not experiencing significant problems, consider adding an extra 50% to the base number. This shouldn't interrupt your users and you still keep some capacity.

In general, the limit you choose should keep your instance safe, not control individual users. Rate limiting is more about protecting Jira from integrations and scripts going haywire, rather than stopping users from getting their work done.

## Adding exemptions

Exemptions are, well, special limits for users who really need to make more requests than others. Any exemptions you choose will take precedence over global settings.



After adding or editing an exemption, you'll see the changes right away, but it takes up to 1 minute to apply the new settings to a user.

List of limited accounts		Exemptions				
You can set individual limits for chosen users by adding them to the list of exemptions. <span style="float: right;">Add exemption</span>						
Start typing to find users <input type="text" value=""/>						
Name	Requests allowed	Per time	Max requests	Actions		
Jie Yan Song	100	1 minute	120	<a href="#">Edit</a>	<a href="#">Remove</a>	
Amar Sundaram	100	1 hour	120	<a href="#">Edit</a>	<a href="#">Remove</a>	
Christopher Palmerson	10000	24 hour	10000	<a href="#">Edit</a>	<a href="#">Remove</a>	

To add an exemption:

1. Open the **Exemptions** tab.
2. Click **Add exemption**.
3. Find the user and choose their new settings.
  - You cant choose groups, but you can select multiple users.
  - The options available here are just the same as in global settings: Allow unlimited requests, Block all requests, or assign custom limit.
4. Click **Save**.

If you want to edit an exemption later, just click **Edit** next to a users name in the **Exemptions** tab.

### Recommended: Add an exemption for anonymous access

Jira sees all anonymous traffic as made by one user: **Anonymous**. If your rate limits are not too high, it might happen that a single user drains the limit assigned to anonymous. Its a good idea to add an exemption for this account with a higher limit, and then observe whether you need to increase it further.

### Identifying users who have been rate limited

When a user is rate limited, theyll know immediately as theyll receive an HTTP 429 error message (too many requests). You can identify users that have been rate limited by opening the *List of limited accounts* tab on the rate limiting settings page. The list shows all users from the whole cluster.

 When a user is rate limited, it takes up to 5 minutes to show it in the table.

List of limited accounts		Exemptions	
The list includes all users that were rate limited across your Jira Data Center in the last 24 hours. <span style="float: right;">Refresh</span>			
Start typing to find users <input type="text"/>			
Name	Number of times limited	Last limited at	Actions
 Jie Yan Song	23,340	07 Jun 2019 10:00am	<a href="#">Add exemption</a>
 Amar Sundaram	14,234	07 Jun 2019 09:59am	<a href="#">Add exemption</a>
 Christopher Palmerson	8,212	07 Jun 2019 09:59am	<a href="#">Add exemption</a>
 Aubrey Graham	2,902	07 Jun 2019 09:59am	<a href="#">Add exemption</a>
 This is a really big named Bot	1,021	07 Jun 2019 09:59am	<a href="#">Add exemption</a>
 Jules Paul Marathon	990	07 Jun 2019 09:59am	<a href="#">Add exemption</a>
 Christopher Jay Breaux	70	07 Jun 2019 09:59am	<a href="#">Add exemption</a>
 Aubrey Graham	5	07 Jun 2019 09:59am	<a href="#">Add exemption</a>
< 1 2 3 >			

### Unusual accounts

Youll recognize the users shown on the list by their name. It might happen, though, that the list will show some unusual accounts, so heres what they mean:

- **Unknown:** Thats a user that has been deleted in Jira. They shouldnt appear on the list for more than 24 hours (as they cant be rate limited anymore), but you might see them in the list of exemptions. Just delete any settings for them, they dont need rate limiting anymore.
- **Anonymous:** This entry gathers all requests that werent made from an authenticated account. Since one user can easily use the limit for anonymous access, it might be a good idea to add an exemption for anonymous traffic and give it a higher limit.

### Adding limited requests to the log file

You can also view information about rate limited users and requests in the Jira log file. This is useful if you want to get more details about the URLs that requests targeted or originated from.

To add limited requests to the log file:

1. Go to **Administration > System > Logging and profiling**.
2. Click **Configure logging level for another package**.
3. Set the package name to:

```
com.atlassian.ratelimiting.internal.requesthandler.logging
```

4. Set the logging level to *DEBUG*, and click **Add**.
5. Every rate limited requests will now appear in the Jira log file:

```
[INFO] [talledLocalContainer] 2019-11-27 16:21:01,610 http-nio-2990-exec-6 DEBUG [internal.requesthandler.logging.RateLimitedRequestLogger] User [admin] has been rate limited for URL [http://localhost:2990/jira/rest/rate-limiting/latest/admin/rate-limit/settings]
[INFO] [talledLocalContainer] 2019-11-27 16:24:29,014 http-nio-2990-exec-5 DEBUG [internal.requesthandler.logging.RateLimitedRequestLogger] User [admin] has been rate limited for URL [http://localhost:2990/jira/rest/rate-limiting/latest/admin/rate-limit/settings], pre-auth
```

## Getting rate limited users perspective

When users make **authenticated** requests, they'll see rate limiting headers in the response. These headers are added to every response, not just when you're rate limited.

Header	Description
X-RateLimit-Limit	The max. number of requests (tokens) you can ever have. New tokens won't be added to your bucket after reaching this limit. Your admin configures this as <i>Max requests</i> .
X-RateLimit-Remaining	The remaining number of tokens. This is what you have and can use right now.
X-RateLimit-Interval-Seconds	The time interval in seconds. You get a batch of new tokens every such time interval.
X-RateLimit-FillRate	The number of tokens you get every time interval. Your admin configures this as <i>Requests allowed</i> .
retry-after	How long you need to wait until you get new tokens.

When you're rate limited and your request doesn't go through, you'll see the HTTP 429 error message (too many requests). You can use these headers to adjust scripts and automations to your limits, making them send requests at a reasonable frequency.

## Allowlisting URLs and external applications

### Allowlisting URLs and resources

We've also added a way to allowlist whole URLs and resources on your Jira instance. This should be used as a quick fix for something that gets rate limited, but shouldn't.

For example, a Marketplace app added some new API to Jira. The app itself is used from the UI, so it shouldn't be limited, but it might happen that Jira sees this traffic as external and applies the rate limit. In this case, you could disable the app or increase the rate limit, but this brings additional complications.

To work around issues like that, you can allowlist the whole resource added by the app so it works without any limits.

1. Go to **Administration > System > General configuration**.
2. Click **Advanced settings**.

3. Find the `com.atlassian.ratelimiting.whitelisted-url-patterns` property, and enter your URLs as a comma-separated list, for example:

```
/**/rest/applinks/**,**/rest/capabilities,**/rest/someapi
```

For more info on how to create URL patterns, see [AntPathMatcher: URL patterns](#).

### Allowlisting external applications

You can also allowlist consumer keys, which lets you remove rate limits for external applications integrated through AppLinks.

 Getting the consumer key looks differently for Atlassian cloud products. If you want to remove rate limits for cloud, see [Removing rate limits for Atlassian cloud products](#).

1. Find the consumer key of your application.
  - a. Go to **Administration > Applications > Application links**.
  - b. Find your application, and click **Edit**.
  - c. Open **Incoming Authentication**, and copy the Consumer Key.
2. Allowlist the consumer key.
  - a. Go to **Administration > System > General configuration**.
  - b. Click **Advanced settings**.
  - c. Enter the consumer key as the value of `com.atlassian.ratelimiting.whitelisted-oauth-consumers`. You can enter multiple consumer keys as a comma-separated list.

After entering the consumer key, the traffic coming from the related application will no longer be limited.

### Adjusting your code for rate limiting

We've created a set of strategies you can apply in your code (scripts, integrations, apps) so it works with rate limits, whatever they are. For more info, see [Adjusting your code for rate limiting](#).

# Adjusting your code for rate limiting

Whether its a script, integration, or app youreusing if its making external REST API requests, it will be affected by rate limiting. Until now, you could send an unlimited number of REST API requests to retrieve data from Jira, so were guessing you havent put any restrictions on your code. When admins enable rate limiting in Jira, theres a chance your requests will get limited eventually, so we want to help you prepare for that.

## Before you begin

To better understand the strategies weve described here, its good to have some some basic knowledge about rate limiting in Jira. When in doubt, head to [Improving instance stability with rate limiting](#) and have a look at the first paragraph.

## Quick reference

- **Success:** When your request is successful, youll get a 2xx code.
- **Error:** When your request fails, youll get a 4xx code. If youre rate limited, it will be 429 (too many requests).

The following HTTP headers are added to every **authenticated** request affected by rate limiting:

Header	Description
X-RateLimit-Limit	The max number of requests (tokens) you can have. New tokens wont be added to your bucket after reaching this limit.
X-RateLimit-Remaining	The remaining number of tokens. This value is as accurate as it can be at the time of making a request, but it might not always be correct.
X-RateLimit-Interval-Seconds	The time interval in seconds. You get a batch of new tokens every time interval.
X-RateLimit-FillRate	The number of tokens you get every time interval.
retry-after	How long you need to wait until you get new tokens. If you still have tokens left, it shows 0; this means you can make more requests right away.

## Strategies

Weve created a set of strategies you can apply to your code so that it works with rate limits. From very specific to more universal, these reference strategies will give you a base, which you can further refine to make an implementation that works best for you.

### 1. Exponential backoff

This strategy is the most universal and the least complex to implement. Its not expecting HTTP headers or any information specific to a rate limiting system, so the same code will work for the whole Atlassian suite, and most likely non-Atlassian products, too. The essence of using it is observing whether youre already limited (wait and retry, until requests go through again) or not (just keep sending requests until youre limited).

+ Universal, works with any rate limiting system.

+ Doesnt require too much knowledge about limits or a rate limiting system.

- High impact on a Jira instance because of concurrency. Were assuming most active users will send requests whenever theyre available. This window will be similar for all users, making spikes in Jira performance. The same applies to threads most will either be busy at the same time or idle.

- Unpredictable. If you need to make a few critical requests, you cant be sure all of them will be successful.

## Summary of this strategy

Here's the high-level overview of how to adjust your code:

1. **Active:** Make requests until you encounter a 429. Keep concurrency to a minimum to know exactly when you reached your rate limit.
2. **Timeout:** After you received a 429, start the timeout. Set it to 1 second for starters. It's a good idea to wait longer than your chosen timeout up to 50%.
3. **Retry:** After the timeout has passed, make requests again:
  - a. Success: If you get a 2xx message, go back to step 1 and make more requests.
  - b. Limited: If you get a 429 message, go back to step 2 and double the initial timeout. You can stop once you reach a certain threshold, like 20 minutes, if that's enough to make your requests work.

With this strategy, you'll deplete tokens as quickly as possible, and then make subsequent requests to actively monitor the rate limiting status on the server side. It guarantees you'll get a 429 if your rate is above the limits.

## 2. Specific timed backoff

This strategy is a bit more specific, as it's using the `retry-after` header. We're considering this header an industry standard and plan to use it across the Atlassian suite, so you can still be sure the same code will work for Bitbucket and Jira, Server and Cloud, etc. This strategy makes sure that you will not be limited, because you'll know exactly how long you need to wait before you're allowed to make new requests.

**+** Universal, works with any rate limiting system within the Atlassian suite (and other products using `retry-after`) Bitbucket and Jira, Server and Cloud, etc.

**+** Doesn't require too much knowledge about limits or a rate limiting system.

**-** High impact on a Jira instance because of concurrency. We're assuming most active users will send requests whenever they're available. This window will be similar for all users, making spikes in Jira performance. The same applies to threads most will either be busy at the same time or idle.

## Summary of this strategy

Here's the high-level overview of how to adjust your code:

1. **Active:** Make requests and observe the `retry-after` response header, which shows the number of seconds you need to wait to get new tokens. Keep concurrency level to minimum to know exactly when the rate limit kicks in.
  - a. Success: If the header says 0, you can make more requests right away.
  - b. Limited: If the header has a number greater than 0, for example 5, you need to wait that number of seconds.
2. **Timeout:** If the header is anything above 0, start the timeout with the number of seconds specified in the header. Consider increasing the timeout by a random fraction, up to 20%.
3. **Retry:** After the timeout specified in the header has passed, go back to step 1 and make more requests.

With this strategy, you'll deplete tokens as quickly as possible, and then pause until you get new tokens. You should never hit a 429 if your code is the only agent depleting tokens and sends requests synchronously.

## 3. Rate adjustment

This strategy is very specific and expects particular response headers, so it's most likely to work for Jira Data Center only. When making requests, you'll observe headers returned by the server (number of tokens, fill rate, time interval) and adjust your code specifically to the number of tokens you have and can use.

**+** It can have the least performance impact on a Jira instance, if used optimally.

**+** Highly recommended, especially for integrations that require high-volume traffic.

**+** Safe, as you can easily predict that all requests that must go through will in fact go through. It also allows for a great deal of customization.

⊖ Very specific, depends on specific headers and rate limiting system.

### Summary of this strategy

Here's the high-level overview of how to adjust your code:

1. **Active:** Make requests and observe all response headers.
2. **Adjust:** With every request, recalculate the rate based on the following headers:
  - `x-ratelimit-interval-seconds`: The time interval in seconds. You get a batch of new tokens every time interval.
  - `x-ratelimit-fillrate`: The number of tokens you get every time interval.
  - `retry-after`: The number of seconds you need to wait for new tokens. Make sure that your rate assumes waiting longer than this value.
3. **Retry:** If you encounter a 429, which shouldn't happen if you used the headers correctly, you need to further adjust your code so it doesn't happen again. You can use the `retry-after` header to make sure that you only make requests when the tokens are available.

### Customizing your code

Depending on your needs, this strategy helps you to:

By following the headers, you should know how many tokens you have, when you will get the new ones, and in what number. The most useful headers here are `x-ratelimit-interval-seconds` and `x-ratelimit-fillrate`, which show the number of tokens available every time interval. They help you choose the perfect frequency of making your requests.

You can wait to perform complex operations until you're sure you have enough tokens to make all the consecutive requests you need to make. This allows you to reduce the risk of leaving the system in an inconsistent state, for example when your task requires 4 requests, but it turns out you can only make 2. The most useful headers are `x-ratelimit-remaining` and `x-ratelimit-interval-seconds`, which show how many tokens you have right now and how long you need to wait for the new ones.

With all the information returned by the headers, you can create more strategies that work best for you, or mix the ones we've described here. For example:

- If you're making requests once a day, you can focus on the max requests you can accumulate (`x-ratelimit-limit`), or lean towards the remaining number of tokens if a particular action in Jira triggers your app to make requests (`x-ratelimit-remaining`).
- If your script needs to work both for Jira Data Center and some other application, use all headers for Jira and focus on the universal `retry-after` or request codes if the app detects different software.

# Jira application cookies

This page lists cookies stored in Jira application users' browsers which are generated by Jira itself. This page does not list cookies that may originate from 3rd-party Jira plugins.

## On this page:

- [Authentication cookies](#)
- [Other Jira cookies](#)

## Authentication cookies

Jira uses [Seraph](#), an open source framework, for HTTP cookie authentication. Jira uses two types of cookies for user authentication:

- The JSESSIONID cookie is created by the application server and used for session tracking purposes. This cookie contains a random string and the cookie expires at the end of every session or when the browser is closed.
- The 'remember my login' cookie (aka the 'remember me' cookie), `seraph.rememberme.cookie`, is generated by Jira when the user selects the **Remember my login on this computer** checkbox on the login page.

 You can read about cookies on the [Wikipedia page about HTTP cookies](#).

## The 'remember my login' cookie

The 'remember my login' cookie, `seraph.rememberme.cookie`, is a long-lived HTTP cookie. This cookie can be used to authenticate an unauthenticated session. Jira generates this cookie when the user selects the **Remember my login on this computer** checkbox on the login page.

### Cookie key and contents

By default, the cookie key is `seraph.rememberme.cookie`, which is defined by the `login.cookie.key` parameter in the `<jira-application-dir>/WEB-INF/classes/seraph-config.xml` file of your [Jira installation directory](#).

The cookie contains a unique identifier plus a securely-generated random string (i.e. token). This token is generated by Jira and is also stored for the user in the Jira database.

### Use of cookie for authentication

When a user requests a web page, if the request is not already authenticated via session-based authentication or otherwise, Jira will match the 'remember my login' cookie (if present) against the token (also if present), which is stored for the user in the Jira database.

If the token in the cookie matches the token stored in the database and the cookie has not expired, the user is authenticated.

### Life of 'remember my login' cookies

You can configure the maximum age of the cookie. To do that you will need to modify the `<jira-application-dir>/WEB-INF/classes/seraph-config.xml` file of your [Jira installation directory](#) and insert the following lines below the other `init-param` elements:

```
<init-param>
  <param-name>autologin.cookie.age</param-name>
  <param-value>2147483</param-value> <!-- The value of ~25 days in seconds -->
</init-param>
```

 **Maximum value**

The value provided in the code snippet above is the maximum you can use for the 'remember my login' cookies. Anything higher will result in errors. [See this bug](#)

## Other Jira cookies

There are several cookies that Jira uses for a variety of other purposes, such as to enhance Jira's security and to store basic presentation and browser capability states, including the type of search view that was last used and various other presentation states. Jira users' authentication details are not stored by these cookies.

Cookie key	Purpose	Cookie contents	Expiry
<b>atlassian.xsrf.token</b>	Helps prevent XSRF attacks. Ensures that during a user's session, browser requests sent to a Jira server originated from that Jira server. For more information about XSRF checking by Jira, see <a href="#">Form Token Checking</a> on the Atlassian Developers site.	Your Jira server's Server ID, a securely-generated random string (i.e. token) and a flag indicating whether or not the user was logged in at the time the token was generated.	At the end of every session or when the browser is closed.
<b>jira.issue.navigator.type</b>	Tracks which type of search view was last used (i.e. simple or advanced searching).	A string indicating the state of your last search view.	Approximately 10 years from the date it is set or was last updated.
<b>AJS.conglomerate.cookie</b>	Tracks which general tabs were last used (e.g. in Jira's plugin manager) or expansion elements were last opened or closed.	One or more key-value strings which indicate the states of your last general tab views or expansion elements.	One year from the date it is set or was last updated.
<b>UNSTOPPABLE_BROWSER_WARNING</b>	Acknowledges that the user has read a message displayed by Jira indicating that the user's browser is not supported by Jira.	A string which indicates that the user has clicked a button acknowledging they have read the message stating they are using an unsupported browser.	At the end of every session or when the browser is closed.
<b>AJS.thisPage</b>	Indicates that the user's browser does not support local storage. This relates to a mechanism used by Jira to store field information in search views when the user clicks their browser's back button.	A string which indicates that the user's browser does not support local storage.	At the end of every session or when the browser is closed.

# Preventing security attacks

This page provides guidelines which, to the best of our knowledge, will help prevent security attacks on your Jira installation.

## Use strong passwords

### Administrators should use strong passwords

All your Jira administrators, Jira system administrators and administrators of all Atlassian applications should have strong passwords. Ask your administrators to update their passwords to strong passwords.

Do not use passwords that are dictionary words. Use mixed-case letters, numbers and symbols for your administrator passwords and make sure they are sufficiently long (e.g. 14 characters). We encourage you to refer to the [Strong Password Generator](#) for guidelines on selecting passwords.

Using strong passwords greatly increases the time required by an attacker to retrieve your passwords by brute force, making such an attack impractical.

### Administrators should have different passwords for different systems

As well as choosing a strong password, administrators should have *different* strong passwords for different systems. This will reduce the impact the attacker can have if they do manage to obtain administrator credentials on one of your systems.

## Apply Jira security patches

Apply the patches found in any security advisories that we release for your version of Jira. These patches protect Jira from recently detected privilege escalation and XSS vulnerabilities.

## Protect against brute force attack

You can also actively protect your systems against repeated unsuccessful login attempts, known as "brute force" login attacks.

### Enable brute force login protection on your Web server

It is possible to also enable brute force login protection on your web server by detecting repeated authentication failures in application logs. Once repeated login failures have been detected, you can set up an automated system to ban access to your web server from that particular IP address.

For more information on how to configure an automated approach to this kind of login prevention, refer to [Using Fail2Ban to limit login attempts](#).

## Restrict network access to administrative sections of applications

An Atlassian application's administration interface is a critical part of the application; anyone with access to it can potentially compromise not only the application instance but the entire machine. As well as limiting access to only users who really need it, and using strong passwords, you should consider limiting access to it to certain machines on the network.

For more information on how to implement Apache blocking rules to restrict access to administrative or sensitive actions in:

- Jira, refer to [Using Apache to limit access to the Jira administration interface](#)
- Confluence, refer to [Using Apache to limit access to the Confluence administration interface](#)

You can use a similar approach to protecting all Atlassian applications.

### On this page:

- [Use strong passwords](#)
- [Apply Jira security patches](#)
- [Protect against brute force attack](#)
- [Restrict network access to administrative sections of applications](#)
- [Restrict file system access by the application server](#)

## Restrict file system access by the application server

The application server (e.g. Tomcat) runs as a process on the system. This process is run by a particular user and inherits the file system rights of that particular user. By restricting the directories that can be written to by the application server user, you can limit unnecessary exposure of your file system to the application.

For example, ensure that only the following directories can be written to by Jira's application server:

- The following subdirectories of your [Jira installation directory](#):
  - logs
  - temp
  - work
- Your [Jira home directory](#)

# Using the Jira application configuration tool

The Jira application configuration tool is an application that offers server-level Jira configuration through a convenient GUI. This tool allows you to do the following:

- [Configure your Jira home directory](#)
- [Configure your database connection](#)
- [Tune your database connection](#)
- [Configure the webserver](#), including the TCP ports that Jira runs through and SSL configuration.

## On this page:

- [Before you begin](#)
- [Starting the Jira configuration tool](#)
- [Configuring the Jira home directory](#)
- [Configuring the database connection](#)
- [Configuring Jira's web server](#)
- [Tuning Jira's database connections](#)

## Before you begin

- The Jira configuration tool requires a Java platform to be installed and configured on your operating system. If you need to install a Java platform to run this tool, we recommend using a [Java platform supported by Jira](#) refer to [Jira requirements](#) for details.
- If you have a console-only connection to your Jira server, you will need to perform these server-level configurations manually.
- Whenever you configure or reconfigure Jira's server-level settings using this tool, Jira **must be restarted** so it can recognize these changes.

## Starting the Jira configuration tool

To use the Jira configuration tool, you must set the `JAVA_HOME` environment variable. If it has not been set already, follow the instructions from [Installing Java](#).

- **Windows:** Open a command prompt and run `config.bat` in the `bin` sub-directory of the [Jira installation directory](#).
- **Linux/Unix:** Open a console and execute `config.sh` in the `bin` sub-directory of the [Jira installation directory](#).

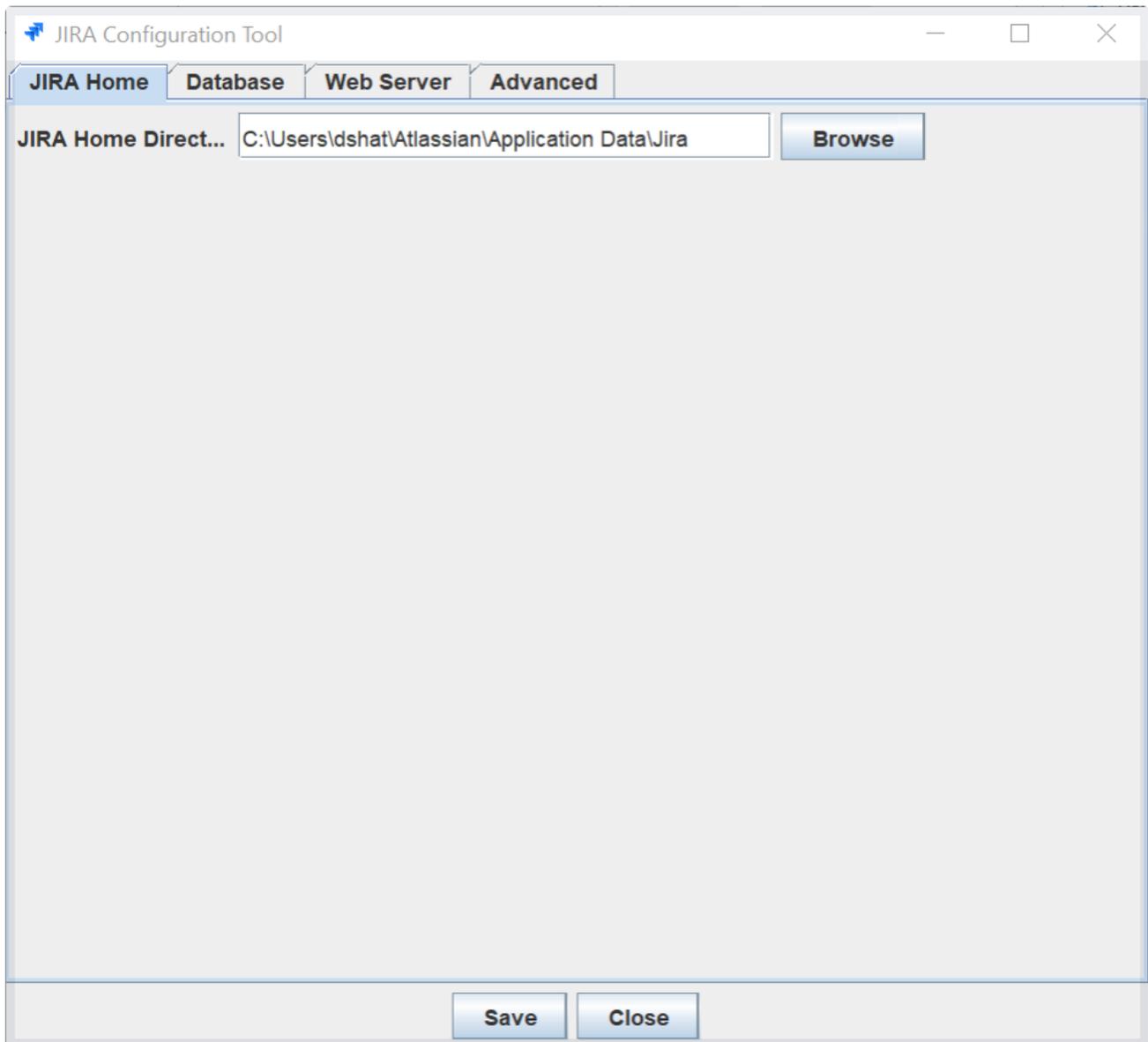
 This command might fail with the error as described in [Unable to Start Jira applications Config Tool due to No X11 DISPLAY variable was set error](#). If it happens, refer to this article for the workaround.

The Jira configuration tool can be run with a graphical user interface or via a command-line interface using the `-c` or `--console` argument. The following sections show the graphical user interface, but the functionality is the same regardless of the interface.

## Configuring the Jira home directory

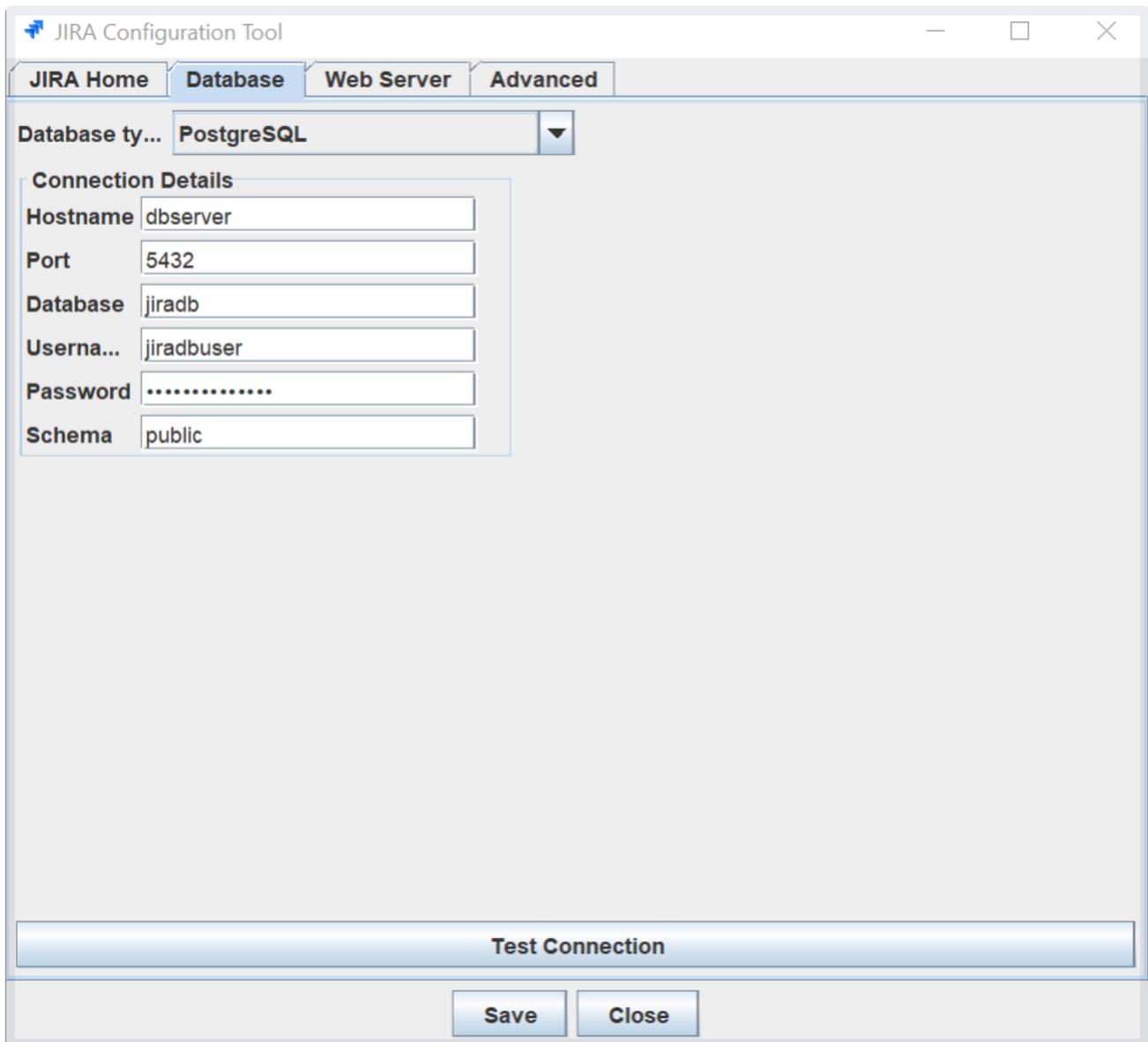
Your [Jira home directory](#) allows you to set the folder that Jira uses to store its various data files.

1. Open the **Jira home** tab.
2. In the Jira home directory field, type the full file path into the text field, or select the **Browse** button to browse for the location of your [Jira home directory](#).
3. Click **Save**. Your changes are saved to the `jira-application.properties` file located in the `<jira-application-dir>` subdirectory of your [Jira application installation directory](#). For more information, see [Setting your Jira home directory](#).



### Configuring the database connection

To configure Jira's database connection by using the Jira configuration tool, follow the appropriate procedure for your [database type](#).



The screenshot shows the JIRA Configuration Tool window with the 'Database' tab selected. The 'Database type' is set to 'PostgreSQL'. The 'Connection Details' section contains the following fields:

Field	Value
Hostname	dbserver
Port	5432
Database	jiradb
Username	jiradbuser
Password	.....
Schema	public

At the bottom of the configuration area is a 'Test Connection' button. Below the configuration area are 'Save' and 'Close' buttons.

## Configuring Jira's web server

You can also use the Jira configuration tool to configure Jira's webserver, specifically the TCP ports and the SSL configuration. To do it, follow the corresponding instructions:

- [Changing Jira's TCP ports](#)
- [Running Jira applications over SSL or HTTPS](#)



### Known issue

When you're adding a new connection, like an SSL one, you'll need to manually add properties that handle special characters to the `server.xml` file. See [this article](#) for more details.

The screenshot shows the 'JIRA Configuration Tool' window with the 'Advanced' tab selected. The 'Web Server' sub-tab is active, displaying the following configuration fields:

- Control Port: 8005
- Profile: HTTPs only
- HTTP Port: 8080
- HTTPs Port: 80
- Keystore Path: C:\Program Files (x86)\Java\jre7\lib\security\cacerts (with a 'Browse' button)
- Keystore Passwo...: changeit
- Key Alias: alias

At the bottom of the configuration area, there is a button labeled 'Check Certificate in Key Store'. Below this, there are 'Save' and 'Close' buttons.

### Tuning Jira's database connections

You can tune all the connections on the **Advanced** tab. For more information about the functionality of the **Advanced** tab, see [Tuning database connections](#).

The screenshot shows the 'JIRA Configuration Tool' window with the 'Advanced' tab selected. The window has a title bar with a minimize button, a maximize button, and a close button. Below the title bar are four tabs: 'JIRA Home', 'Database', 'Web Server', and 'Advanced'. The 'Advanced' tab is active and contains two main sections: 'Scalability and Performance' and 'Eviction Policy'. Each section contains a list of settings with checkboxes and input fields.

Section	Setting	Value / State
Scalability and Performance	<input checked="" type="checkbox"/> Maximum Size	30
	<input type="checkbox"/> Maximum Idle	
	<input type="checkbox"/> Minimum Idle/Size	
	<input type="checkbox"/> Initial Size	
	<input type="checkbox"/> Maximum Wait Time	
	<input type="checkbox"/> Pool Statements	<input type="checkbox"/>
	<input type="checkbox"/> Maximum Open Statemen..	
Eviction Policy	<input type="checkbox"/> Validation Query	
	<input type="checkbox"/> Validation Query Timeout	
	<input type="checkbox"/> Test On Borrow	<input type="checkbox"/>
	<input type="checkbox"/> Test On Return	<input type="checkbox"/>
	<input type="checkbox"/> Test While Idle	<input type="checkbox"/>
	<input checked="" type="checkbox"/> Time Between Eviction Runs	5000
	<input checked="" type="checkbox"/> Minimum Evictable Idle Time	4000
	<input type="checkbox"/> Remove Abandoned On Borrow	<input type="checkbox"/>
	<input type="checkbox"/> Remove Abandoned On Maintenan..	<input type="checkbox"/>
<input type="checkbox"/> Remove Abandoned Timeout		

At the bottom of the window are two buttons: 'Save' and 'Close'.

# Running Jira applications as a Windows service

For long-term use, Jira should be configured to automatically restart when the operating system restarts. For Windows servers, this means configuring Jira to run as a **Windows service**.

Running Jira as a Windows service has other advantages. When started manually, a console window opens, and there is a risk of someone accidentally shutting down Jira by closing this window. Also, the Jira logs are properly managed by the Windows service (found in `logs\stdout*`. Login your [Jira installation directory](#), and rotated daily).

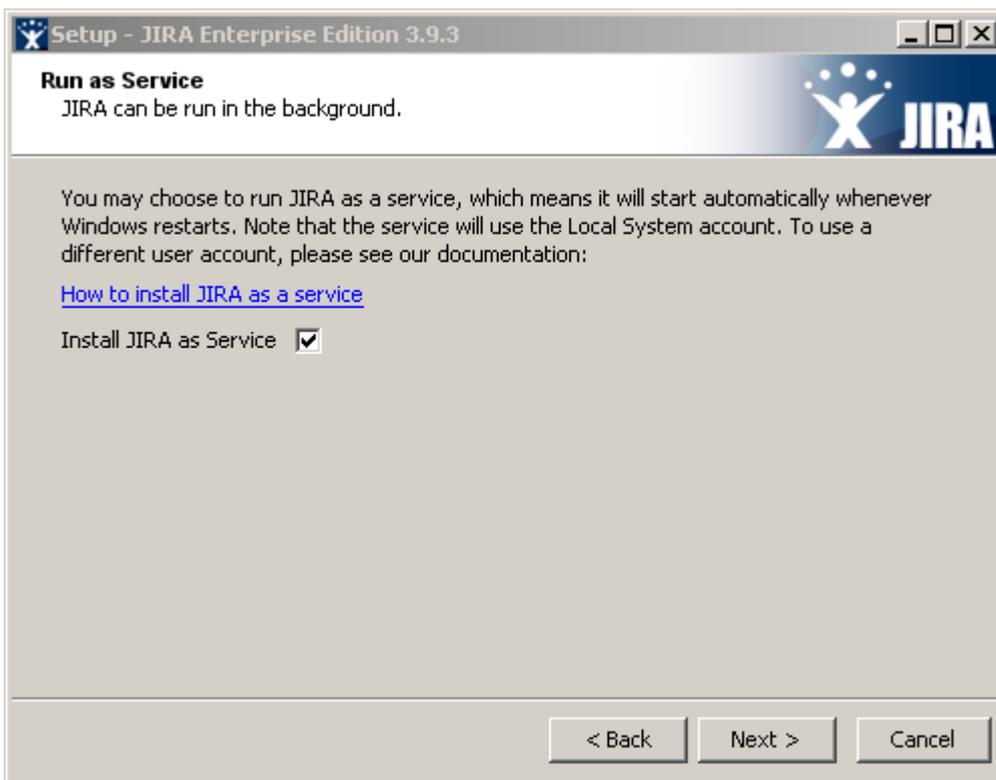
There are two ways to install Jira as a service: via the installer, and manually.

## On this page:

- [Installing as a service with the installer](#)
- [Removing the Jira service](#)
- [Changing the Windows user that the Jira service uses](#)
- [Specifying the startup order of multiple services](#)
- [Locating the name of a service](#)
- [Troubleshooting](#)

## Installing as a service with the installer

The easiest way to get Jira installed as a Windows service is by clicking the **'Install Jira as Service'** checkbox when running the [Windows Installer](#):



 You will need full Administrator rights on your Windows operating system for this installation process to complete successfully.

## Manually setting up Jirato run as a service

You can still set up Jirato run as a service, if any of the following situations apply to you:

- You did not use the [Windows Installer](#).

- You used the Windows Installer, but did not initially install Jira as a service.

#### ⓘ Please note:

- On any Windows operating system with User Account Control (UAC), such as Windows Vista or Windows 7, you must either disable UAC or run 'cmd.exe' as an administrator (e.g. by right-clicking on 'cmd.exe' and selecting "Run as administrator") in order to execute the script in the procedure below. If UAC is enabled, simply logging in to Windows with an Administrator account will not be sufficient.

#### To set up Jira to run as a service:

- Open a Command Prompt.
- Change directory ('cd') to the [Jira application installation directory](#) and then into this directory's 'bin' subdirectory.
  - ⚠ If a directory in the path has spaces (e.g. 'C:\Program Files\..'), please convert it to its eight-character equivalent (e.g. 'C:\Progra~1\..').
- Ensure the **JAVA\_HOME** variable is set to the root of your Java platform's installation directory.
  - ⓘ To find out the current value of the **JAVA\_HOME** variable, enter `echo %JAVA_HOME%` at the command prompt.
- To ensure that Jira won't fail to start due to a bad parameter, add the `XX: +IgnoreUnrecognizedVMOptions` parameter at Jira startup. This parameter allows to bypass any bad arguments being passed to Java.
- Run the following command:
- `service.bat install Jira`

Here is a screenshot of the process:

```

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Denis>cd "c:\Program Files\atlassian-jira-enterprise-3.3-standalone"
C:\Program Files\atlassian-jira-enterprise-3.3-standalone>cd bin
C:\Program Files\atlassian-jira-enterprise-3.3-standalone\bin>echo %JAVA_HOME%
C:\j2sdk1.4.2_02
C:\Program Files\atlassian-jira-enterprise-3.3-standalone\bin>service.bat install JIRA
Installing the service 'JIRA' ...
Using CATALINA_HOME:   C:\Program Files\atlassian-jira-enterprise-3.3-standalone
Using JAVA_HOME:       C:\j2sdk1.4.2_02
Using JVM:              C:\j2sdk1.4.2_02\jre\bin\server\jvm.dll
The service 'JIRA' has been installed.
C:\Program Files\atlassian-jira-enterprise-3.3-standalone\bin>tomcat5 //US//JIRA --Startup auto
C:\Program Files\atlassian-jira-enterprise-3.3-standalone\bin>tomcat5 //US//JIRA --JvmMx 512
C:\Program Files\atlassian-jira-enterprise-3.3-standalone\bin>
  
```

Jira should now be set up to run as a service.

- In addition, to have the Jira service start automatically when the operating system starts, run:

```
tomcat8 //US//JIRA --Startup auto
```

The Jira service will automatically start up the next time the operating system reboots. The Jira service can be manually started with the command '**net start JIRA**' and stopped with '**net stop JIRA**'.

- ⓘ To see what parameters the JIRA Core service is starting with, go to **Start -> Run** and run '`regedt32.exe`' and then:
  - \* For Windows 32 bit edition navigate to HKEY\_LOCAL\_MACHINE -> SOFTWARE -> Apache Software Foundation -> Procrun 2.0 -> JIRA<time stamp>
  - \* For Windows 64 bit edition navigate to HKEY\_LOCAL\_MACHINE -> SOFTWARE -> Wow6432Node -> Apache Software Foundation -> Procrun 2.0 -> JIRA<time stamp>

### 8. Additional Jirasetup options (optional):

- To increase the maximum memory Jiracan use (the default will already be 256MB), run:

```
tomcat8 //US//service_name --JvmMx 512
```

where **service\_name** is the name of your Jiraservice, e.g. JIRA123487934298.

- If you are running Jira and Confluence in the same JVM, increase the MaxPermSize size to 128 MB:

```
tomcat8 //US//service_name ++JvmOptions="-XX:MaxPermSize=128m"
```

where **service\_name** is the name of your Jiraservice, e.g. JIRA123487934298.

- Occasionally, it may be useful to view Jira's Garbage Collection information. This is especially true when investigating memory issues. To turn on the Verbose GC (garbage collection) logging, execute the following command in the command prompt:

```
tomcat8 //US//service_name ++JvmOptions="-Xloggc:path\to\logs\atlassian-gc.log"
```

where **service\_name** is the name of your Jiraservice, e.g. JIRA123487934298.

The path (denoted by **path\to**) refers to the directory in which Jira is currently installed. For example:

```
tomcat8 //US//service_name ++JvmOptions="-Xloggc:c:\jira\logs\atlassian-gc.log"
```

where **service\_name** is the name of your Jiraservice, e.g. JIRA123487934298.

 See [the Tomcat documentation](#) for further service options.

## Removing the Jiraservice

If Jira was installed through the Windows installer, go to the '**Control Panel**' in Windows, click '**Add or Remove Programs**' and remove Jira. This will remove the service too.

If you installed the service manually (see above) it can be uninstalled with:

```
service.bat remove JIRA
```

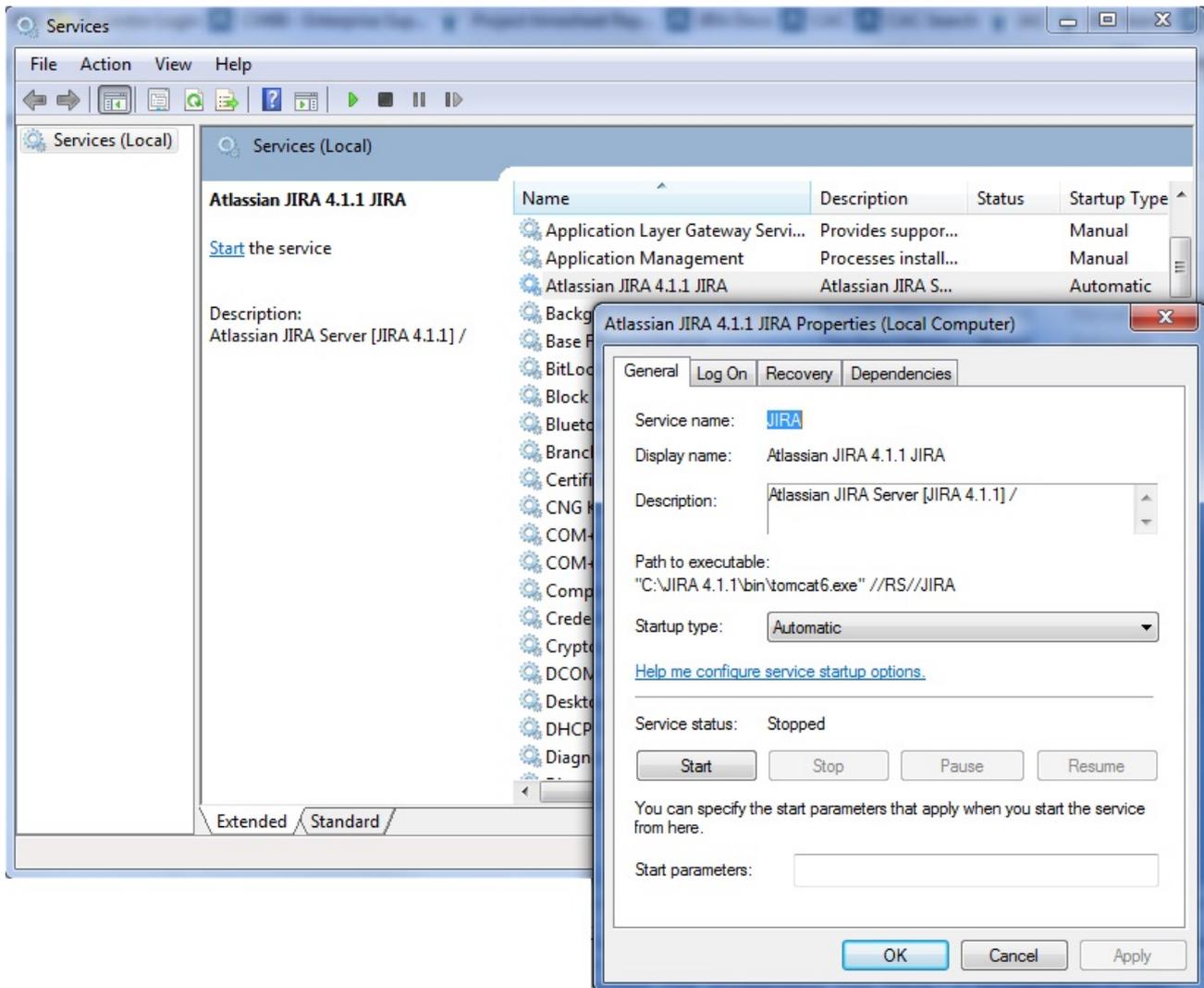
Alternatively, if the above does not work, use `tomcat8 //DS//JIRA`.

## Changing the Windows user that the Jiraservice uses

If you are using mapped network drives for Jira's backup directory, attachments directory, index directory or the `%CATALINA_HOME%\*` directory, you need to ensure that Jiracan write to these drives. That is, these directories all need to be writeable by the user which the Jiraservice is running as. This may mean that you need to change the Windows user that the Jiraserver uses.

 Note that you must also specify these network drives by UNC and not letter mappings, e.g. `\\backupserve r\jira` not `z:\jira`

To change the Windows user that the Jiraservice uses, navigate to the service in Windows, i.e. '**Control Panel**' -> '**Administrative Tools**' -> '**Services**'. Locate the 'Atlassian Jira' service, right-click and view the 'Preferences'.



Go to the '**Log On**' tab and change the user as desired.

## Specifying the startup order of multiple services

If you have services that depend on each other, it is important that they are started in the correct order. Common examples include:

- If you are running both JIRA and [Crowd](#), it is important to start Crowd first, so that Crowd is running before people try to login to Jira.
- If the database Jira connects to is hosted on the same server as Jira, and is started via a Windows service, the Jira service will only start successfully if the database service has already started first.

To set up start up dependency rules, open a command prompt and enter the following command:  
`C:\Documents and Settings\Developer>sc config [JIRA service] depend=[database service]`

*Please note the space character after 'depend='.*

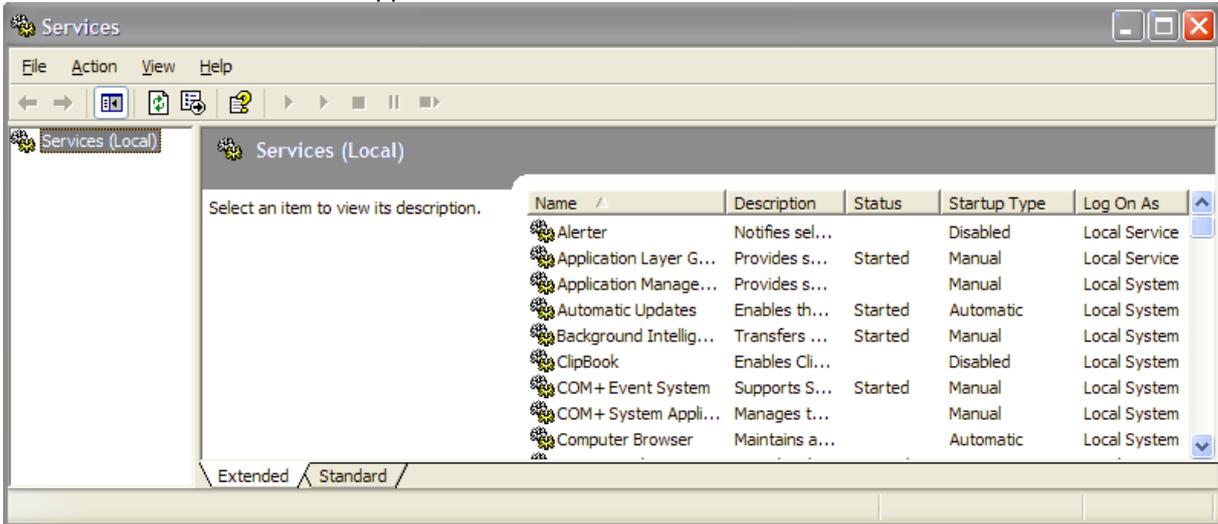
- **[JIRA service]** is the name of the Jira service you are running, e.g. JIRA051007111904.
- **[database service]** is the name of the database service you are running, e.g. MSSQLSERVER.

If you wish, you can also set up dependency rules by editing the system registry. Please see <http://support.microsoft.com/kb/193888> for details on how to do this.

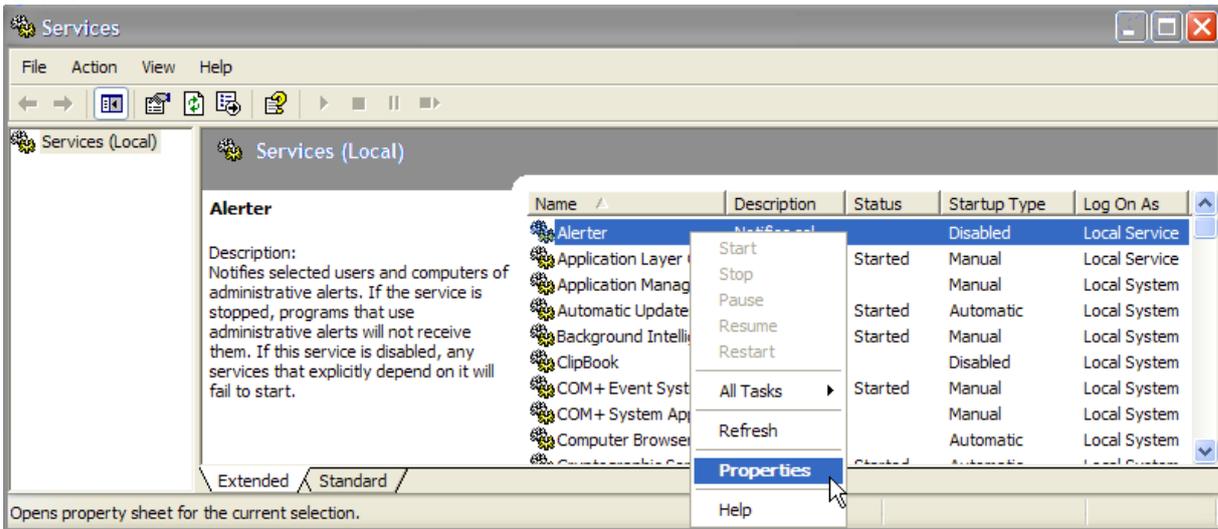
## Locating the name of a service

If you do not know the exact name of your Jiraservice or your database service, you can find out what they are by following the steps below:

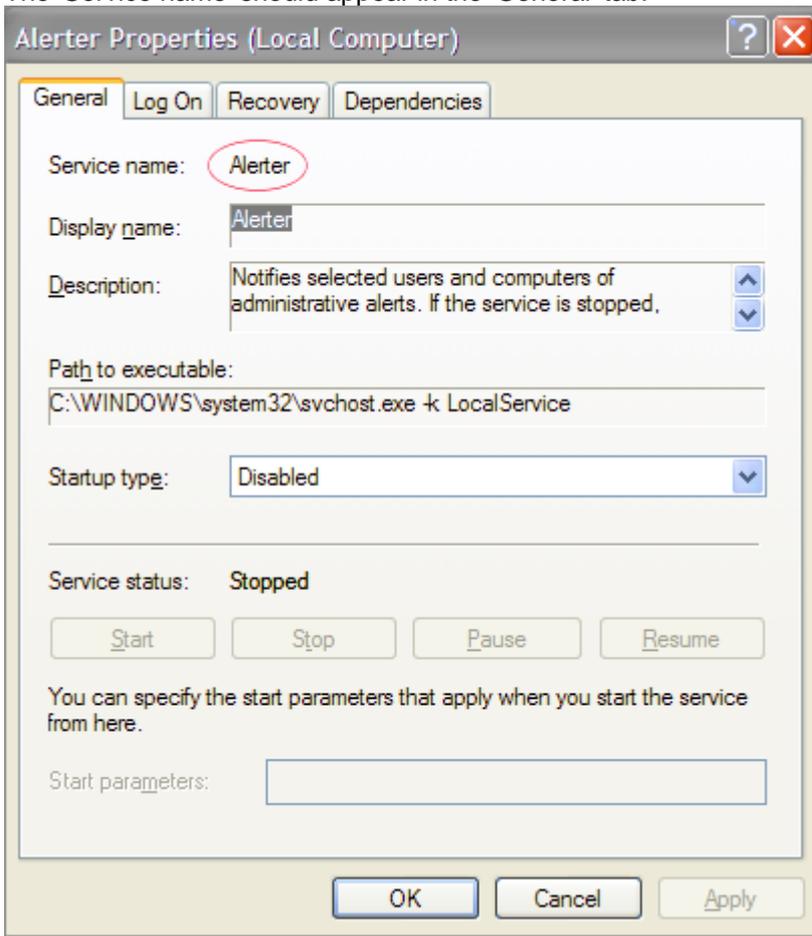
1. Navigate to '**Control Panel**' > '**Administrative Tools**' > '**Services**'.
2. The 'Services' window should appear:



3. Right-click on the service you wish to find out the name of, and select '**Properties**' from the popup menu:



4. The 'Service name' should appear in the 'General' tab:



## Troubleshooting

- Java 6 is not supported by Jira 6.0 and later. Problems may occur when trying to setup Jira to run as a Windows service with JDK 1.6. The problem is due to failure to locate "MSVCR71.DLL", which can be found in %JAVA\_HOME%/bin. There are two options to resolve this problem:
  - Add %JAVA\_HOME%/bin to PATH, then restart the Jira server.
  - Copy MSVCR71.DLL to system path, C:\WINDOWS\SYSTEM32 or C:\WINNT\SYSTEM32
- Take note of the username that the service is running as, and be sure to modify the /temp and /work directories in your install directory so that this user has read and write permissions.
- You cannot run Jira as a service on a 64-bit operating system if you require allocating more than 1.5 GB of memory, due to 32-bit JDK memory limitations and 64-bit JDK/Tomcat service issues.

# Tuning garbage collection (GC)

Jira applications are robust applications that rarely require in-depth garbage collection (GC) tuning. However, on large-scale installations, GC tuning can improve the performance of Jira applications. Analysis of GC logs can also assist in troubleshooting performance problems. This page lists basic recommendations that improve garbage collection, and is an entry point to the advanced GC Tuning Guide. The recommendations are based on Support's successful experiences with customers with large Jira instances.

You should get yourself familiar with garbage collection if:

- Jira has high memory or CPU consumption, or
- Jira is performing slowly or has occasional outages

## Tuning garbage collection

### Checking the GC performance

The first step towards improving GC is actually measuring the GC performance of the JVM to see the size of the problem. You can do it by viewing and analyzing the GC logs. The logs indicate when the JVM is collecting garbage, how long this process takes, and how much memory has been freed. Starting from Jira 7.4, GC logs are generated automatically, and you can find them in `<installation-directory>/logs`. [GC Viewer](#) is a great tool for analyzing GC logs. To enable garbage collection in earlier Jira versions, see [Using garbage collection logs](#).

### Garbage Collectors

Don't use the Concurrent Mark Sweep (CMS) Collector unless otherwise advised by Atlassian Support. It requires extensive manual tuning and testing, and is likely to result in degraded performance. Instead, we recommend that you use Garbage First Garbage Collector (G1GC), especially if your heap is larger than 6 GB.

### Heap size

Start with the smallest heap possible, and increase it in 512 MB or 1 GB allotments when you experience OutOfMemory errors or run into any memory-related problems. Do not increase the heap further than required, as this will result in longer garbage collection. After garbage collection, the memory used should be 1/3 the size of the total heap, and that's your goal.

### Old tuning parameters

On every full garbage collection, the JVM will resize the allocations of Eden, Survivor, etc., based on the throughput it is actually seeing. It will tune itself based on the real world data of the objects that are being created and collected. Most of the time simply allowing JVM to tune itself will give you better performance.

If you have added JVM parameters in the past and are experiencing difficulties with GC now, we'd recommend you remove all GC related parameters, unless you added them to solve a specific problem, and they did in fact solve that problem. You should also consider re-benchmarking now to ensure that they are still solving that problem, and are not causing you any other issues.

### VM resources

If you run Jira on a VM, check that it's not using the swap file. If it does, during the garbage collection, the JVM has to load the objects from the swap file into memory to clean them, and this can cause significantly longer GC pauses. Instead of using swapping, ballooning and bursting, allocate adequate memory to the VM.

### Manual tuning

If you find you are still experiencing difficulties with GC after following these recommendations and you would like to see if you can tune the JVM better to improve performance, we recommend following the instructions in our [Garbage Collection \(GC\) Tuning Guide](#). This document will take you through the process of choosing performance goals (throughput/footprint/latency), and how to tune for these goals.

# Jira Data Center documentation

Data Center is our self-managed edition of Jira built for enterprises. It provides the deployment flexibility and administrative control you need to manage mission-critical Jira sites.

## Server and Data Center features comparison

Want to see what's included with a Data Center license? Head to the [Jira Server and Data Center feature comparison](#).

 You can [purchase a Data Center license](#) or create an evaluation license at [my.atlassian.com](https://my.atlassian.com)

## Data Center deployment options

You can deploy Jira Data Center in two ways:



### Non-clustered (single node)

Run Jira Data Center on a single node, just like a Server installation. This option doesn't require any changes to your infrastructure, but it does allow you to take advantage of Data Center-only features. Quick and easy. [Learn more](#)



### Clustered

Run Jira Data Center in a cluster with multiple nodes, and a load balancer to distribute traffic. Clustering is designed for large, or mission-critical, Jira instances, allowing you to provide high availability, and maintain performance as you scale. [Learn more](#)

## Get started

- [Whitepaper: An admin's guide to getting started with Jira Data Center](#)

### Non-clustered

- [Learn about non-clustered architecture and requirements](#)
- [Install Jira Data Center from scratch](#)
- [Migrate from Server to Data Center](#)
- [Move from non-clustered to clustered Data Center](#)

### Clustered

- [Learn about clustered architecture and requirements](#)
- [Set up a Data Center cluster](#)
- [Add or remove application nodes](#)
- [Revert to a non-clustered Data Center installation](#)

# Running Jira Data Center on a single node

You can run Jira Data Center on a single node, just like a Server installation. This is useful if you don't need extra capacity or other benefits that clustering provides.

## Benefits of running a non-clustered Data Center deployment

There are a range of reasons you may choose a single node Data Center. Some of the benefits include:

- **Keeping your existing infrastructure**  
Running on a single node means that you can upgrade from Server to Data Center without adding to your infrastructure. In most cases, moving to Data Center will be as simple as updating your license.
- **Accessing Data Center-only features**  
Your Data Center license unlocks a suite of additional security, compliance, and administration features to help you easily manage enterprise-grade Jira instance like SAML single sign-on, project and issue archiving, rate limiting, and more. [See the complete list](#)

**i** Unlike a Server installation, non-clustered Data Center deployments are cluster-compatible, which means you can still enable and configure clustering whenever you're ready to scale. [Learn more about setting up a cluster](#)

## Architecture

The image below shows a typical configuration:



As you can see, Jira Data Center deployed on a single node looks just as a Server installation, and consists of:

- Jira Data Center, running on a single node
- A database that Jira read and writes to

## Requirements

Non-clustered Data Center deployments follow the same minimum requirements as a Server installation. Check our Jira [Installation requirements](#) for more details.

## App compatibility

The process for installing Marketplace apps (also known as add-ons or plugins) in Jira Data Center is the same as for Server. You won't have to stop Jira to install or update an app.

The Atlassian Marketplace indicates apps that are compatible with Jira Data Center. [Learn more about Data Center approved apps](#)

Ready to get started?

Deploying Data Center on a single node will be the same as deploying a Server installation, just with a different license.

- If you're installing from scratch, head to [Installing Jira applications](#) to learn about all the ways in which you can install Jira.
- If you already have a Server instance, head to [Migrate from Server to Data Center](#).

# Moving back to Server

If you no longer need Data Center, you can go back to a Server installation. This will be as easy as updating your license, but be aware that losing the Data Center-only features might affect your Jira instance.

## How will the Data Center-only features be affected?

After moving from Data Center to Server, you will lose features exclusive to Data Center. Here's the summary of how this will affect your current Jira instance:

Feature (non-clustered)	Result
<b>Project and issue archiving</b> Archive projects and issues once they're no longer needed.	<ul style="list-style-type: none"><li>Archived projects and issues will be restored to their original state</li><li>You will no longer have access to Archived projects and Archived issues pages</li><li>You'll need to perform a full re-index to make the restored projects and issues searchable</li></ul>
<b>SAML single sign-on</b> Use a SAML identity provider for authentication and single-sign on.	<ul style="list-style-type: none"><li>SAML login won't work i.e. the users will be redirected to a login screen always</li><li>Users who had a password on their Atlassian account before SAML single sign-on was enabled will use that to log in</li><li>Users who joined after SAML single sign-on was enabled will need to reset their password for their Atlassian account when they next log in</li></ul>
<b>Custom fields optimizer</b> Scan and manage custom fields easily to optimize your JiraSoftware instance's performance.	<ul style="list-style-type: none"><li>No longer available</li><li>Custom fields already optimized in Data Center won't be affected</li></ul>
<b>Content Delivery network (CDN) support</b> Improve geo-performance for distributed teams.	<ul style="list-style-type: none"><li>No longer available</li><li>Any API calls related to the feature will be blocked</li><li>If you've already configured your CDN, then switch back to a Server license, the CDN <b>will continue to work</b> until your Jira instance is restarted</li></ul>
<b>Rate limiting</b> Control how many external REST API requests users and automations can make and how often they can make them.	<ul style="list-style-type: none"><li>No longer available</li><li>Any configured limits that were set up before will become irrelevant and your users will have unlimited external REST API requests the moment the license is downgraded</li></ul>

## Before you begin

- This guide assumes that you're using a non-clustered (single node) Jira Data Center. If you have a cluster of nodes, go to [Moving to a non-clustered Data Center](#) first.
- We recommend completing this process in a staging environment, and running a set of functional tests, integration tests, and performance tests, before making these changes in production.

## Moving back to Server

### 1. Get a license for Jira Server

You can [purchase a Server license](#) or create an evaluation license at [my.atlassian.com](https://my.atlassian.com).

### 2. Apply the license

1. Go to **Administration** () > **Applications > Versions and licenses**.
2. Update the license for each of your applications.

After updating the license, your Jira instance will become a Server one, and you'll immediately lose access to Data Center-only features.

# Running Jira Data Center in a cluster

Jira Data Center allows you to run a cluster of multiple Jira nodes, providing high availability, scalable capacity, and performance and scale. We'll tell you about the benefits, and give you an overview of what you'll need to run Jira in a clustered environment.

**Ready to get started?** See [Set up a Jira Data Center cluster](#).

## Benefits of clustering

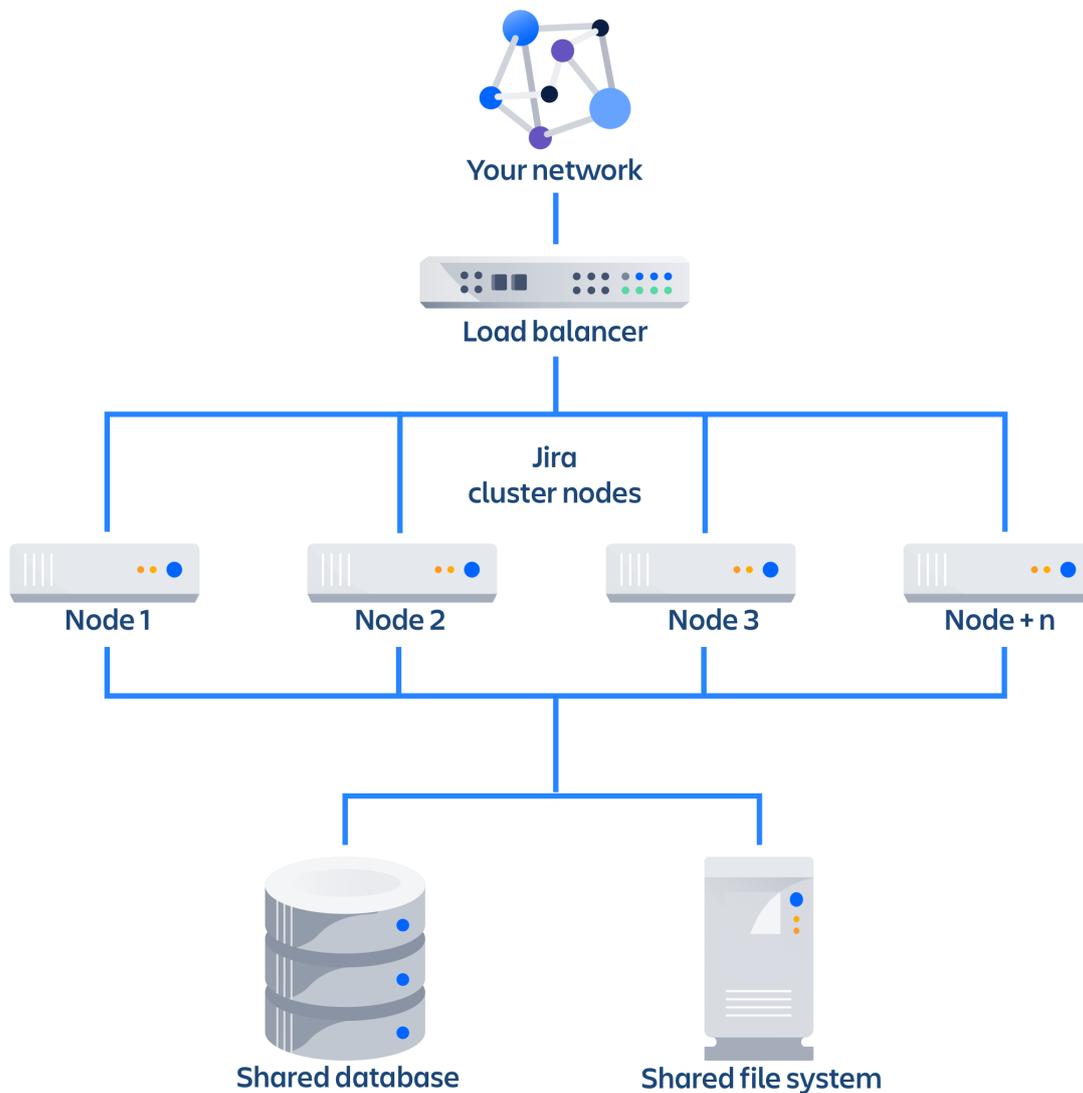
Clustering is designed for enterprises with large or mission-critical Data Center deployments that require continuous uptime, instant scalability, and performance under high load.

Here are some of the benefits:

- **High availability and failover**  
If one node in your cluster goes down, the other take on the load, ensuring your users have uninterrupted access to Jira.
- **Performance and scale**  
Each node added to your cluster increases concurrent user capacity, and improves response time as user activity grows.
- **Instant scalability**  
Add new nodes to your cluster without downtime or additional licensing fees. Indexes and apps are automatically synced.

## Architecture

The image below shows a typical configuration:



As you can see, a Jira Data Center cluster consists of:

- Multiple identical application nodes running Jira Data Center.
- A load balancer to distribute traffic to all of your application nodes.
- A shared filesystem that stores attachments, and other shared files.
- A database that all nodes read and write to.

All application nodes are active and process requests. A user will access the same Jira node for all requests until their session times out, they log out, or a node is removed from the cluster.

#### Learn more

Your Data Center license is based on the number of users in your cluster, rather than the number of nodes. This means you can scale your environment without additional licensing fees for new servers or CPU.

You can monitor the available license seats in the Versions & licenses page in the admin console.

If you wanted to automate this process (for example to send alerts when you are nearing full allocation) you can use the REST API.

Your Jira license determines which features and infrastructure choices are available. Head to [Jira Server and Data Center feature comparison](#) for a full run down of the differences between a Server license and a Data Center license.

To run Jira in a cluster, you'll need an additional home directory, known as the shared home.

Each Jira node has a local home that contains logs, caches, Lucene indexes and configuration files. Everything else is stored in the shared home, which is accessible to each Jira node in the cluster.

Here's a summary of what is found in the local home and shared home:

Local home	Shared home
<ul style="list-style-type: none"> <li>• logs</li> <li>• caches</li> <li>• Lucene indexes</li> <li>• configuration files</li> <li>• plugins</li> </ul>	<ul style="list-style-type: none"> <li>• attachments</li> <li>• avatars / profile pictures</li> <li>• icons</li> <li>• export files</li> <li>• import files</li> <li>• plugins</li> <li>• cluster status and synchronization data</li> </ul>

In Jira Data Center, cache modifications are replicated between the nodes to keep all of them in sync. We're using asynchronous cache replication, which means that modifications aren't replicated immediately after they occur, but are added to local queues (each node has local queues for every other node), and then replicated based on their order in the queue. With this approach, we can improve the scalability of the cluster, reduce the amount of cache inconsistencies, and separate the replication itself from any cache modifications, which simplifies and speeds up the whole process.

For more info, see [Jira Data Center cache replication](#).

Each individual Jira application node stores its own full copy of the index. A journal service keeps each index in sync.

When you first set up your cluster, you will copy the local home directory, including the indexes, from the first node to each new node.

When adding a new Jira node to an existing cluster, you will copy the local home directory of an existing node to the new node. When you start the new node, Jira will check if the index is current, and if not, request a recovery snapshot of the index from either the shared home directory, or a running node (with a matching build number) and extract it into the index directory before continuing the start up process. If the snapshot can't be generated or is not received by the new node in time, existing index files will be removed, and Jira will perform a full re-index.

If a Jira node is disconnected from the cluster for a short amount of time (hours), it will be able to use the journal service to bring its copy of the index up-to-date when it rejoins the cluster. If a node is down for a significant amount of time (days), its Lucene index will have become stale, and it will request a recovery snapshot from an existing node as part of the node startup process.

If you suspect there is a problem with the index on all nodes, you can temporarily disable index recovery on one node, rebuild the index on that node, then copy the new index over to each remaining node.

For more info, see [Jira Data Center search indexing](#).

Where an action must only run on one node, for example a scheduled job or sending daily email notifications, Jira uses a cluster lock to ensure the action is only performed on one node.

Cluster locks are acquired and then released by a node. To make sure that a cluster lock doesn't block the whole cluster if one of the nodes goes offline, we use a heartbeat mechanism that regularly checks if the node that acquired the lock is still active. This mechanism can release the lock, if needed.

When configuring your cluster nodes you can either supply the IP address of each cluster node, or a multicast address.

**If you're using multicast:**

Jira will broadcast a join request on the multicast network address. Jira must be able to open a UDP port on this multicast address, or it won't be able to find the other cluster nodes. Once the nodes are discovered, each responds with a unicast (normal) IP address and port where it can be contacted for cache updates. Jira must be able to open a UDP port for regular communication with the other nodes.

A multicast address can be auto-generated from the cluster name, or you can enter your own, during the set-up of the first node.

## Infrastructure and requirements

The choice of hardware and infrastructure is up to you. Below are some areas to think about when planning your hardware and infrastructure requirements.

### Running Jira Data Center on Kubernetes

If you plan to run Jira Data Center on Kubernetes, you can use our Helm charts. For more information, see [Running Jira Data Center on a Kubernetes cluster](#).

### Deploying Jira Data Center on AWS and Azure

If you plan to run Jira Data Center on AWS or Azure, you can use our templates to deploy the whole infrastructure. You'll get your Jira Data Center nodes, database and storage all configured and ready to use in minutes. For more info, see the following resources:

- [Getting started with Jira Data Center on AWS](#)
- [Getting started with Jira Data Center on Azure](#)

### Server requirements

You should not run additional applications (other than core operating system services) on the same servers as Jira. Running Jira, Confluence and Bamboo on a dedicated Atlassian software server works well for small installations but is discouraged when running at scale.

Jira Data Center can be run successfully on virtual machines.

### Cluster nodes requirements

Each node does not need to be identical, but for consistent performance we recommend they are as close as possible. All cluster nodes must:

- be located in the same data center, or region (for AWS and Azure)
- run the same Jira version
- have the same OS, Java and application server version
- have the same memory configuration (both the JVM and the physical memory) (recommended)
- be configured with the same time zone (and keep the current time synchronized). Using ntpd or a similar service is a good way to ensure this.

You must ensure the clocks on your nodes don't diverge, as it can result in a range of problems with your cluster.

### How many nodes?

Your Data Center license does not restrict the number of nodes in your cluster. The right number of nodes depends on the size and shape of your Jira instance, and the size of your nodes. See our [Jira Data Center size profiles](#) guide for help sizing your instance. In general, we recommend starting small and growing as you need.

### Memory requirements

We recommend that each Jira node has a minimum of 8GB RAM. This would be sufficient for a single Server instance with a small number of projects (up to 100) with 1,000 to 5,000 issues in total and about 100-200 users.

To get an idea on how large and complex your Jira instance is, see [Jira Data Center size profiles](#).

The maximum heap (-Xmx) for the Jira application is set in the `setenv.sh` or `setenv.bat` file. The default should be increased for Data Center. We recommend keeping the minimum (Xms) and maximum (Xmx) heap the same value.

You can also check the details of our public Jira Data Center instances. See [Jira Data Center sample deployment](#).

## Database

You should ensure your intended database is listed in the current [Supported platforms](#). The load on an average cluster solution is higher than on a standalone installation, so it is crucial to use the a supported database.

You must also use a supported database driver, which should be listed in supported platforms linked above. For more detailed instructions on connecting Jira to a database, see [Connecting Jira applications to a database](#).

## Additional requirements for database high availability

Running Jira Data Center in a cluster removes the application server as a single point of failure. You can also do this for the database through the following supported configurations:

- [Amazon RDS Multi-AZ](#): this database setup features a primary database that replicates to a standby in a different availability zone. If the primary goes down, the standby takes its place.
- [Amazon PostgreSQL-Compatible Aurora](#): this is a cluster featuring a database node replicating to one or more readers (preferably in a different availability zone). If the writer goes down, Aurora will promote one of the writers to take its place.

The [AWS Quick Start deployment option](#) allows you to deploy Jira Data Center with either one, from scratch. If you want to set up an Amazon Aurora cluster with an existing Jira Data Center instance, refer to [Configuring Jira Data Center to work with Amazon Aurora](#).

## Shared home and storage requirements

All Jira cluster nodes must have access to a shared directory in the same path. NFS and SMB/CIFS shares are supported as the locations of the shared directory. As this directory will contain large amount of data (including attachments and backups) it should be generously sized, and you should have a plan for how to increase the available disk space when required.

## Load balancers

We suggest using the load balancer you are most familiar with. The load balancer needs to support session affinity. If you're deploying on AWS you'll need to use an Application Load Balancer (ALB).

Here are some recommendations when configuring your load balancer:

- Queue requests at the load balancer. By making sure the maximum number requests served to a node does not exceed the total number of http threads that Tomcat can accept, you can avoid overwhelming a node with more requests than it can handle. You can check the `maxThreads` in `<install-directory>/conf/server.xml`.
- Don't replay failed idempotent requests on other nodes, as this can propagate problems across all your nodes very quickly.
- Using *least connections* as the load balancing method, rather than *round robin*, can better balance the load when a node joins the cluster or rejoins after being removed.

Many load balancers require a URL to constantly check the health of their backends in order to automatically remove them from the pool. It's important to use a stable and fast URL for this, but lightweight enough to not consume unnecessary resources. The following URL returns Jira's status and can be used for this purpose.

URL	Expected content	Expected HTTP status
-----	------------------	----------------------

http://<jiraurl>/status		{"state":"RUNNING"}	200 OK
HTTP status code	Response entity	Description	
200	{"state":"RUNNING"}	Running normally	
500	{"state":"ERROR"}	An error state	
503	{"state":"STARTING"}	Application is starting	
503	{"state":"STOPPING"}	Application is stopping	
200	{"state":"FIRST_RUN"}	Application is running for the first time and has not yet been configured	
404		Application failed to start up in an unexpected way (the web application failed to deploy)	

Here are some recommendations, when setting up monitoring, that can help a node survive small problems, such as a long GC pause:

- Wait for two consecutive failures before removing a node.
- Allows existing connection to the node to finish, for say 30 seconds, before the node is removed from the pool.

For more info, see [Load balancer configuration options](#) and [Load balancer examples](#).

## Network adapters

Use separate network adapters for communication between servers. Cluster nodes should have a separate physical network (i.e. separate NICs) for inter-server communication. This is the best way to get the cluster to run fast and reliably. Performance problems are likely to occur if you connect cluster nodes via a network that has lots of other data streaming through it.

## App compatibility

The process for installing Marketplace apps (also known as add-ons or plugins) in a Jira cluster is the same as for a standalone installation. You will not need to stop the cluster, or bring down any nodes to install or update an app.

The Atlassian Marketplace indicates apps that are compatible with Jira Data Center. [Learn more about Data Center approved apps](#)

## Ready to get started?

Head to [Set up a Jira Data Center cluster](#) for a step-by-step guide to enabling and configuring your cluster.

# Set up a Jira Data Center cluster

Jira Data Center allows you to run a cluster of multiple Jira nodes, providing high availability, scalable capacity, and performance at scale. This guide walks you through the process of configuring a Data Center cluster on your own infrastructure.

 Not sure if clustering is right for you? Check out [Running Jira Data Center in a cluster](#) for a detailed overview.

## Before you begin

Things you should know about when setting up your Data Center:

See our [Supported platforms](#) page for information on the database, Java, and operating systems you'll be able to use. These requirements are the same for Server and Data Center deployments.

To use Jira Data Center, you must:

- Have a Data Center license (you can [purchase a Data Center license](#) or create an evaluation license at [my.atlassian.com](#))
- Use a [supported](#) external database, operating system and Java version
- Use OAuth authentication if you have [application links](#) to other Atlassian products (such as Confluence)

To run Jira in a cluster, you must also:

- Use a load balancer with session affinity and WebSockets support in front of the Jira cluster. [Load balancer examples](#)
- Have a shared directory accessible to all cluster nodes in the same path (this will be your shared home directory). This must be a separate directory, and not located within the local home or install directory.

Apps extend what your team can do with Atlassian applications, so it's important to make sure that your team can still use their apps after migrating to Data Center. When you switch to Data Center, you'll be required to switch to the Data Center compatible version of your apps, if one is available.

See [Evaluate apps for Data Center migration](#) for more information.

In this guide we'll use the following terminology:

- **Installation directory:** The directory where you installed Jira.
- **Local home directory:** The home or data directory stored locally on each cluster node (if Jira is not running in a cluster, this is simply known as the home directory).
- **Shared home directory:** The directory you created that is accessible to all nodes in the cluster via the same path.

## Set up and configure your cluster

### 1. Install or upgrade your Jira instance

Jira Data Center is available for Jira 7.0, or later. If you're not on this version yet, install or upgrade your Jira instance.

[Jira installation and upgrade guide](#)

### 2. Set up the shared directory

You'll need to create a remote directory that is readable and writable by all nodes in the cluster. There are multiple ways to do this, but the simplest is to use an NFS share.

1. Create a remote directory, accessible by all nodes in the cluster, and name it e.g. `sharedhome`.
2. Stop your Jira instance.

- Copy the following directories from the Jira local home directory to the new shared home directory (some of them may be empty).

- data
- plugins
- logos
- import
- export
- caches
- keys

### 3. Configure your Jira instance to work in a cluster

- In the Jira local home directory, create `cluster.properties` file, with contents as follows:

#### Example cluster.properties file:

```
# This ID must be unique across the cluster
jira.node.id = node1
# The location of the shared home directory for all Jira nodes
jira.shared.home = /data/jira/sharedhome
```

For more information and some additional parameters, see [Cluster.properties file parameters](#).

- For Linux installations:** We recommend that you increase the maximum number of open files. To do that, add the following line to `<jira-install>/bin/setenv.sh`:

```
ulimit -n 16384
```

- Start your instance, and [apply the Data Center license](#).

### 4. Add the first node to the load balancer

The load balancer distributes the traffic between the nodes. If a node stops working, the remaining nodes will take over its workload, and your users won't even notice it.

- Add the first node to the load balancer.
- Restart the node, and then try opening different pages in Jira. If the load balancer is working properly, you should have no problems with accessing Jira.

### 5. Add the remaining nodes to the cluster

The approach to adding the remaining nodes to the cluster varies with the method that was used to install Jira on the first node (either manually from a `.zip` or `.tar.gz` archive or using a `.bin` or `.exe` installer). Follow the steps that correspond to the original installation method.

- Copy the Jira installation and home directories from an existing node to the new node.
- Ensure the new node can read and write to the shared home directory.
- Edit `<home-directory>/cluster.properties` on the new node by providing a unique node ID and an IP address if one was specified.
- Start Jira. It will read the configuration from the shared home directory and start without any extra setup.
- Take a look around the new Jira instance. Ensure that issue creation, search, attachments, and customizations work as expected.
- If everything looks fine, you can configure your load balancer to start routing traffic to the new node. Once you do this, you can make a couple of changes in one Jira instance to see if they're visible in other instances as well. Use the same method to install the same version of Jira on another node in your cluster. During the installation, take note of the locations of the Jira installation and home directory paths.

1. Ensure the new node can read and write to the shared home directory.
2. Start Jira to allow the application to populate the home directory.
3. Open Jira in the browser and make sure that you can see the setup page. If the page appears, the installation was successful and you can close the browser.
4. Stop Jira.
5. Copy `dbconfig.xml` and `cluster.properties` from the Jira home directory on an existing node to the Jira home directory on the new node.
6. Copy `server.xml` from `<installation-directory>/conf` on an existing node to `<installation-directory>/conf` on the new node.
7. Edit `<home-directory>/cluster.properties` on the new node by providing a unique node ID and an IP address if one was specified.
8. If you modified any [important directories and files](#) (for example, `<installation-directory>/bin/setenv.sh` or `<installation-directory>/conf/web.xml`) on an existing node, copy the modified files to the same locations on the new node.
9. If Jira runs over SSL, import the SSL certificates to the local Java truststore on the new node to allow Jira to communicate with itself over its base URL.
10. Start Jira. It will read the configuration from the shared home directory and start without any extra setup.
11. Take a look around the new Jira instance. Ensure that issue creation, search, attachments, and customizations work as expected.
12. If everything looks fine, you can configure your load balancer to start routing traffic to the new node. Once you do this, you can make a couple of changes in one Jira instance to see if they're visible in other instances as well.



While adding your nodes to the cluster, you can check their status by going to  **System > System info**. Your nodes will be listed in the **Cluster nodes** section.

### Cluster.properties file parameters

In addition to the required parameters, the `cluster.properties` file allows you to configure some additional options, mostly related to EhCache.

Parameter	Required	Description/value
<code>jira.node.id</code>	Yes	This unique ID must match the username and the <code>BalancerMember</code> entry in the Apache configuration.
<code>jira.shared.home</code>	Yes	The location of the shared home directory for all Jira nodes.
<code>ehcache.peer.discovery</code>	No	<p>Describes how nodes find each other:</p> <p><code>default</code> Jira will automatically discover nodes (recommended)</p> <p><code>automatic</code> Jira will use the EhCache's multicast discovery. This is the historical method used by EhCache, but it can be difficult to configure, and is not recommended by Atlassian.</p> <p>If you set <code>ehcache.peer.discovery=automatic</code> then you need to set the following parameters:</p> <ul style="list-style-type: none"> <li>• <code>ehcache.multicast.address</code></li> <li>• <code>ehcache.multicast.port</code></li> <li>• <code>ehcache.multicast.timeToLive</code></li> <li>• <code>ehcache.multicast.hostName</code></li> </ul> <p>For more info on these parameters, see <a href="#">Ehcache documentation</a>.</p>

<code>ehcache. listener. hostName</code>	No	The hostname of the current node for cache communication. Jira Data Center will resolve this internally if the parameter isn't set. If you have problems resolving the hostname of the network you can set this parameter.
<code>ehcache. listener.port</code>	No	The port that the node is going to be listening to (default is 40001).  If multiple nodes are on the same host, or if this port is unavailable, you might need to set this parameter manually.
<code>ehcache. object.port</code>	No	The port on which the remote objects bound in the registry receive calls (default is 40011). Make sure you also open this port on your firewall.  If multiple nodes are on the same host, or if this port is unavailable, you might need to set this parameter manually.
<code>ehcache. listener. socketTimeoutM illis</code>	No	By default, this is set to the EhCache default.

# Adding and removing Data Center nodes

You can add and remove nodes from your Data Center cluster at any time. You don't have to worry about extra nodes affecting your license, because it's based on the number of users in the cluster, rather than the number of nodes.

## Adding a node

To add a node:

1. Stop one of your nodes.
2. Copy the installation directory and local home directory from the stopped node to your new node.
3. Edit the `cluster.properties` file in the local home directory and change `jira.node.id` to a new, unique identifier.
4. Start Jira on your new node.  
During the startup process, Jira will recover indexes from a running node to bring the new node up to date.
5. Go to **Administration** (⚙️) > **System** > **Clustering** and check that the new node is visible.

You should only start one node at a time. Starting up multiple nodes simultaneously can cause serious failures.

## Removing a node

To remove a node, stop Jira on that node. You can then remove the installation and local home directory as required.

To see the number of nodes remaining, go to **Administration** (⚙️) > **System** > **Clustering**.

## Scaling up and down in AWS and Azure

If you deployed Jira Data Center on AWS or Azure, your Jira nodes will be in scaling groups. You will add and remove nodes either by changing the minimum and maximum size of each group or using a scaling plan. See the following resources for more info:

- [Administering Jira Data Center on AWS](#)
- [Administering Jira Data Center in Azure](#)

## Moving to non-clustered Data Center

If you no longer need clustering, but still want access to Data Center features, you can go back to a non-clustered (single node) Data Center installation. In these instructions we'll assume that you'll use one of your existing cluster nodes as your new installation. You'll also need to make some infrastructure changes as part of the switch.

### What about the features?

After moving to a non-clustered (single node) Data Center deployment, you will lose capabilities that are based on clustering, such as zero-downtime upgrades, instant scalability, and performance and scale. For the complete list of features, see [Server and Data Center feature comparison](#).

## Before you begin

- We recommend completing this process in a staging environment, and running a set of functional tests, integration tests, and performance tests, before making these changes in production.

### 1. Shut down Jira

Stop all cluster nodes before you proceed.

### 2. Configure your load balancer

Configure your load balancer to redirect traffic away from all Jira nodes, except the node you plan to use for your standalone installation.

If you no longer need your load balancer, you can remove it at this step.

### 3. Move items in the cluster shared home back to local home

- Move the contents of your `<shared home>` directory to the root of your `<local home>` directory. Make sure your attachments directory is moved as part of this step.

Your cluster's shared home directory should now be empty.

### 4. Delete the `cluster.properties` file

- Delete the `cluster.properties` file that you created when setting up your cluster. The file is in the local home directory.

### 5. Start Jira

Restart Jira.



To confirm you're now running a non-clustered installation, go to **Administration > System > Clustering**. The active cluster should no longer appear. Instead, you'll see some general information about clustering in Jira Data Center.

# Performance and scaling

In this section, we present some best practices on how large and growing teams use Jira Software, along with the results of performance and scale tests we run for each release. These results can show you how the latest Jira version compares to the previous one, and how different metrics (projects, issues, custom fields, and so on) can affect your instance.

- [Best practices for scaling Jira Software](#)
- [Performance and scale testing](#)
- [Use a CDN with Atlassian Data Center applications](#)
- [Configure your CDN for Jira Data Center](#)
- [Jira Data Center monitoring](#)

# Best practices for scaling Jira Software

In this guide, we'll look at the best ways to use Jira Software for both large organizations and teams that wish to scale. If you have a large Jira instance with multiple Jira applications, then you'll want to start with [Performance and scale testing](#), which shows the performance of Jira Data Center and how different factors and data sets (issues, users, attachments, etc.) affect it.

This page will provide specific information about scaling the Jira Software application.

## Skip to

- [Board design and usage](#)
- [Cutting up the workload](#)
- [How Atlassian uses Jira Software](#)
- [Client side considerations](#)
- [Atlassian Long Term Support releases](#)
- [Work with our Solution Partners](#)

## Board design and usage

There are a number of things to keep in mind when configuring boards for scale. Not only can you include issues from multiple projects, but you can also exclude issues based on [JQL](#)- this can lead to confusion of what is and is not appearing on the board. Generally it's best to start off thinking about the true goals of this board. Keep in mind, you can create multiple boards (even multiple boards that cover the same issues) to show data differently for different teams.

Whenever you create a new board using the Scum or Kanban presets, a corresponding filter will be created for you as well - this means you can modify the filter to narrow down the issues on the board, removing those you don't need to see and letting you focus on only what you need.

It's important to note that while boards can display up to 2,000 issues, it's not recommended to be viewing that many records on the board at once. Boards that have 500 or more issues to display will take a longer time to render in some browsers (IE8 for example) or on lower powered machines (see the bottom section on Client Side Considerations for more info). Additionally, it can be difficult to find the issues you want in a sea of cards. Additionally, your Jira instance will need to pull up the data about these issues at each refresh - if there's a thousand issues there and 25 people want to load this board at the same time, then there's a bit of work to be done there.

The issue data itself is pulled from the indexes with the new boards, rather than from the database directly, so this should greatly improve the load times you'll see on larger boards. This should also be taken into consideration when scaling a Jira instance.

Since boards also inherit the permissions of the filters they are based upon, it's now possible to create boards visible only to certain groups or roles, to help alleviate the clutter you might otherwise see when trying to find the right board.

## Cutting up the workload

If you find you have an overwhelmingly large board then usually the first place to look is the quick filters. [Quick filters](#) allow you to narrow the board down to a subset of the issues that match both the original filter and the quick filter. Often we've seen large boards with a number of quick filters that could easily be broken down into two or three smaller boards. Start by looking at the quick filters that are used most often and look at what they focus on. Also take a look at filtering out issues that are stale. If an issue has been sitting around for over a year without an update, what kind of value is it providing to anyone? Perhaps setup a second board that collects all the old, out of date issues and occasionally triage these closing them off or updating them (promoting them to the primary board).

JQL allows you to search by a wide variety of aspects of the issue - assignees and reporters (both individuals and groups), issue age, time since last update and of course the historical searches allowing you to see things that at one point in time were in a particular status, assignee etc. By focusing on the particulars that make issues important to you, it will become much easier to get a board with exactly the issues you need.

Once that initial filter is built there are a number of ways to increase focus on the issues you need:

 Although the following items help you focus on your work, they're not recommended if you're already struggling with poor performance. Both swimlanes and card colors complicate the JQL queries, which makes a board load much slower.

- **Swimlanes** allow you to set the order of certain issues based on the JQL you need. For example, you may want to see your issues organized by priority, promoting criticals and blockers to the top. Alternatively you might want to organize it by assignee so you can see who is doing what.
- **Quick Filters** are great for quickly grabbing the issues you need such as your assigned issues, or only the recently updated issues so you can see what's changed since yesterday.
- **Card Colors** allow you to customize the colors of the cards on your board either based on issue type, or arbitrary JQL so you can highlight issues of importance.

## How Atlassian uses Jira Software

Within Atlassian we have quite a number of small teams, but we do have two particularly large teams that have each had a different focus and way of handling issues - as well as dramatically different Jira instances.

Within Support, we run a Kanban style board that we have broken down into smaller individual boards based on the regional teams we have. Since we do not rank issues within Support (instead going on a first in, first out basis after priorities are taken into account) we have decided to change our filter to not Order By Rank, instead we order by a combination of priority and time in status.

In addition to this, we've heavily customized the Swimlanes to promote Escalated tickets, Critical Issues, Enterprise and unassigned issues higher than non-critical or already assigned issues, allowing us to quickly scan the top row for the most important issues all the time. After that we have each individual engineers workload visible which allows us to quickly determine who can take on the next issue.

On the other hand, the Jira Development team has broken themselves into smaller sub teams based on the themes of the work they are doing. This allows them to individually have their own boards on a per team basis, while the Product Managers use an aggregate board of the Stories to see the work being done by each team. By focusing on the bigger picture, they can simplify the board and summarize where everything up to, as well as quickly see which teams have the most stories to work on.

## Client side considerations

Since the new boards are now doing a lot more work client side than they used to, you may find that when you first upgrade that some users on older browsers or lower powered machines are experiencing slower load times. Cutting down the board size or reviewing the browser type and client machine specs will help alleviate these issues.

## Atlassian Long Term Support releases

An Atlassian Long Term Support release is a feature release that gets backported security updates and critical bug fixes during its entire two-year support window. If you can only upgrade once a year, consider upgrading to a Long Term Support release. [Learn more](#)

## Work with our Solution Partners

Especially when your Jira Software has reached a more advanced size and complexity, we would recommend to collaborate with one of our Solution Partners.

We can help connect you with a right Partner. If you're interested for a referral, please [contact us](#).

# Performance and scale testing

With every Jira release, we're publishing a performance and scaling report that compares performance of the current Jira version with the previous one. The report also contains results of how various data dimensions (number of custom fields, issues, projects, and so on) affect Jira, so you can check which of these data dimensions should be limited to have best results when scaling Jira.

 **This report is for Jira 8.20** . If you're looking for other reports, select your version at the top-right.

## Skip to

- [Introduction](#)
- [Determining the scale of a single Jira instance](#)
- [Jira 8.20 performance](#)
- [Testing methodology](#)
- [Jira 8.20 scalability](#)
- [Performance of Advanced Roadmaps in Jira Data Center](#)
- [Further resources](#)

## Introduction

When some Jira administrators think about how to scale Jira, they often focus on the number of issues a single Jira instance can hold. However, the number of issues is not the only factor that determines the scale of a Jira instance. To understand how a large instance may perform, you need to consider multiple factors.

This page explains how Jira performs across different versions and configurations. So whether you are a new Jira evaluator that wants to understand how Jira can scale to your growing needs or you're a seasoned Jira administrator that is interested in taking Jira to the next level, this page is here to help.

There are two main approaches, which can be used in combination to scale Jira across your entire organization:

1. Scale a single Jira instance.
2. Use Jira Data Center which provides Jira clustering.

Here we'll explore techniques to get the most out of Jira that are common to both approaches. For additional information on Jira Data Center and how it can improve performance under concurrent load, please refer to our [Jira Data Center page](#).

## Determining the scale of a single Jira instance

There are multiple factors that may affect Jira's performance in your organization. These factors fall into the following categories (in no particular order):

- Data size
  - The number of issues, comments, and attachments.
  - The number of projects.
  - The number of Jira project attributes, such as custom fields, issue types, and schemes.
  - The number of users registered in Jira and groups.
  - The number of boards, and the number of issues on the board (when you're using Jira Software).
- Usage patterns
  - The number of users concurrently using Jira.
  - The number of concurrent operations.
  - The volume of email notifications.
- Configuration
  - The number of plugins (some of which may have their own memory requirements).
  - The number of workflow step executions (such as Transitions and Post Functions).
  - The number of jobs and scheduled services.

- Deployment environment
  - Jira version used.
  - The server Jira runs on.
  - The database used and connectivity to the database.
  - The operating system, including its file system.
  - JVM configuration.

This page will show how the speed of Jira can be influenced by the size and characteristics of data stored in the database.

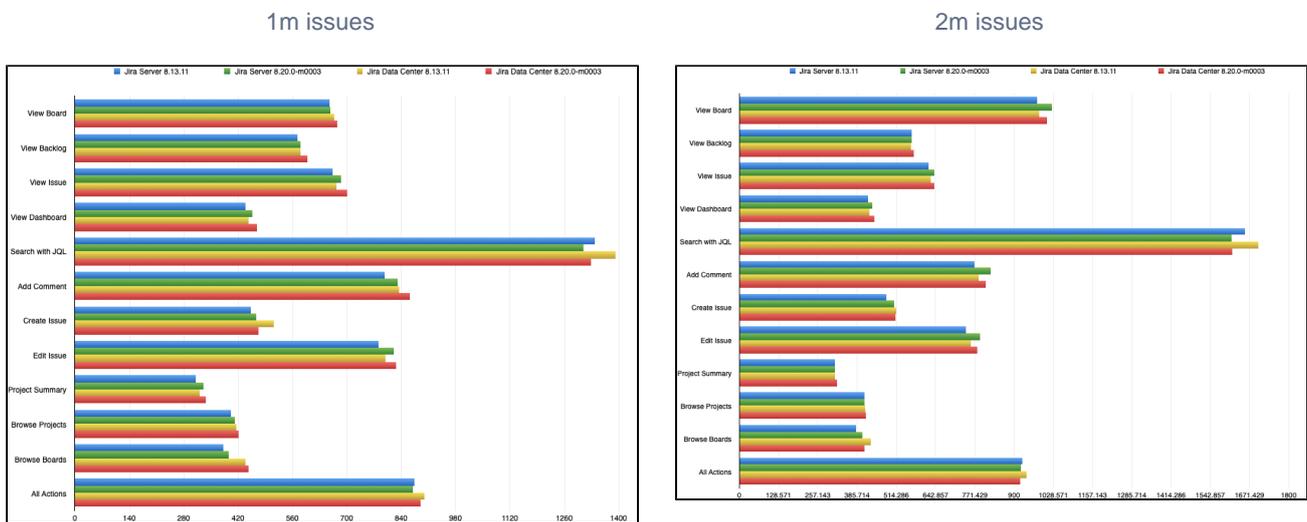
### Jira 8.20performance

Jira 8.20 was not focused solely on performance, however we do aim to provide the same, if not better, performance with each release. In this section, we'll compare Jira 8.13 to Jira 8.20. We ran the same extensive test scenario for both Jira versions. The only difference between the scenarios was the Jira version.

The following chart presents mean response times of individual actions performed in Jira. To check the details of these actions and the Jira instance they were performed in, see [Testing methodology](#).

#### Response times for Jira actions

#### Jira 8.13.11 vs Jira 8.20.0



### Testing methodology

The following sections detail the testing environment, including hardware specification, and methodology we used in our performance tests.

Before we started the test, we needed to determine what size and shape of data set represents a typical large Jira instance.

In order to achieve that, we used our Analytics data to form a picture of our customers' environments and what difficulties they face when scaling Jira in a large organization.

The following table presents rounded values of the 999th percentile of each data dimension. We used these values to generate a sample dataset with random test data in [Jira Data Generator](#).

#### Baseline Jira data set

Jira data dimension	Value
Issues	1,000,000
Projects	1500

Custom Fields	1400
Workflows	450
Attachments	660,000
Comments	2,900,000
Agile Boards	1,450
Users	100,000
Groups	22,500
Security Levels	170
Permissions	200

We chose a mix of actions that would represent a sample of the most common user actions. An "action" in this context is a complete user operation like opening of an Issue in the browser window. The following table details the actions that we included in the scenario for our testing persona, indicating how many times an action is performed in a single test run, where a full set of performance tests involves multiple test runs.

Action name	Description	Number of actions per test run
<b>View Dashboard</b>	Opening the Dashboard page.	10
<b>Create Issue</b>	Submitting a Create Issue dialog.	5
<b>View Issue</b>	Opening an individual issue in a separate browser window.	55
<b>Edit Issue</b>	Editing an issue, submitting the changes, and waiting until they appear in the issue.	5
<b>Add Comment</b>	Adding a comment, and waiting until it appears in the issue.	2

<b>Search with JQL</b>	<p>Performing a search query using JQL in the Issue Navigator interface.</p> <pre>resolved is not empty order by description</pre> <pre>text ~ \"a*\" order by summary</pre> <pre>priority in [priorityname] order by reporter</pre> <pre>project = [projectname] order by status</pre> <pre>assignee = [username] order by project</pre> <pre>reporter was [username] order by description</pre> <pre>project = [projectname] and assignee = [username] order by reporter</pre> <pre>text ~ \"[text]\" order by key</pre>	20
<b>View Board</b>	Opening of Agile Board	10
<b>Browse Projects</b>	Opening of the list of Projects (available under <i>Projects &gt; View All Projects</i> menu)	5
<b>Browse Boards</b>	Opening of the list of Agile Boards (available under <i>Agile &gt; Manage Boards</i> menu)	2
<b>Project Summary</b>	Opening the <b>Project Summary</b> page and waiting for the metadata column to appear	5
<b>View Backlog</b>	Opening the <b>Backlog</b> page	10
<b>All Actions</b>	A weighted arithmetic mean of all actions performed during a single test run. The weight of each data point corresponds to the request count per action per test run.	-

The performance tests were all run on a set of AWS EC2 instances, deployed in the `eu-central-1` region. For each test, the entire environment was reset and rebuilt, and then each test started with some idle cycles to warm up instance caches. Below, you can check the details of the environments used for Jira Server and Jira Data Center, as well as the specifications of the EC2 instances.

To run the tests, we used 20 scripted browsers and measured the time taken to perform the actions. Each browser was scripted to perform a random action from a predefined list of actions and immediately move on to the next action (ie. zero think time). Please note that it resulted in each browser performing substantially more tasks than would be possible by a real user and you should not equate the number of browsers to represent the number of real-world concurrent users.

Each test was run for 20 minutes, after which statistics were collected.

Jira Server	Jira Data Center
The environment with Jira Server consisted of: <ul style="list-style-type: none"> <li>• 1 Jira node</li> <li>• Database on a separate node</li> <li>• Load generator on a separate node</li> </ul>	The environment with Jira Data Center consisted of: <ul style="list-style-type: none"> <li>• 2 Jira nodes</li> <li>• Database on a separate node</li> <li>• Load generator on a separate node</li> <li>• Shared home directory on a separate node</li> <li>• Load balancer (AWS ELB HTTP load balancer)</li> </ul>

Jira			
Hardware		Software	
<b>EC2 type:</b>	c5d.9xlarge (see <a href="#">EC2 types</a> ) Jira Server: 1 node Jira Data Center: 2 nodes	<b>Operating system:</b>	Ubuntu 16.04 LTS
<b>CPU:</b>	3.0 GHz Intel Xeon Platinum 8000-series	<b>Java platform:</b>	Java 1.8.0
<b>CPU cores:</b>	36	<b>Java options:</b>	8 GB heap
<b>Memory:</b>	72 GB		
<b>Disk:</b>	900 GB NVMe SSD		

Database			
Hardware		Software	
<b>EC2 type:</b>	c4.8xlarge (see <a href="#">EC2 types</a> )	<b>Database:</b>	MySQL 5.5
<b>CPU:</b>	Intel Xeon E5-2666 v3 (Haswell)	<b>Operating system:</b>	Ubuntu 16.04 LTS
<b>CPU cores:</b>	36		
<b>Memory:</b>	60 GB		

<b>Disk:</b>	Jira Server: AWS EBS 100 GB gp2 Jira Data Center: AWS EBS 60 GB gp2		
Load generator			
Hardware		Software	
<b>EC2 type:</b>	c4.8xlarge (see <a href="#">EC2 types</a> )	<b>Operating system:</b>	Ubuntu 16.04 LTS
<b>CPU:</b>	Intel Xeon E5-2666 v3 (Haswell)	<b>Browser:</b>	Google Chrome 62
<b>CPU cores:</b>	36	<b>Automation script:</b>	Chromedriver 2.33
<b>Memory:</b>	60 GB		WebDriver 3.4.0
<b>Disk:</b>	AWS EBS 30 GB gp2		Java JDK 8u131

## Jira 8.20scalability

Jira's flexibility causes tremendous diversity in our customer's configurations. Analytics data shows that nearly every customer dataset displays a unique characteristic. Different Jira instances grow in different proportions of each data dimension. Frequently, a few dimensions become significantly bigger than the others. In one case, the issue count may grow rapidly, while the project count remains constant. In another case, the custom field count may be huge, while the issue count is small.

Many organizations have their own unique processes and needs. Jira's ability to support these various use cases explains the dataset diversity. However, each data dimension can influence Jira's speed. This influence is often not constant nor linear.

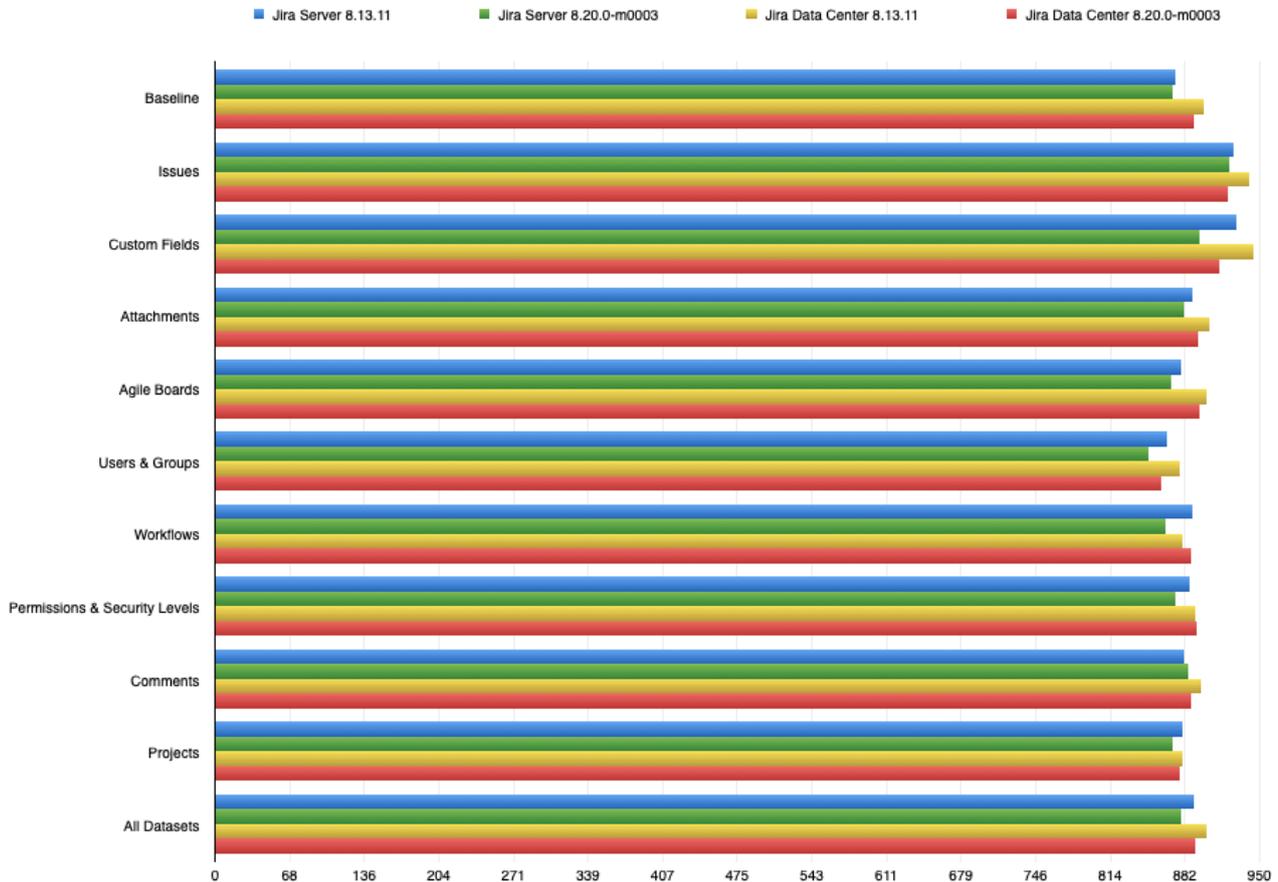
In order to provide individual Jira instance users with an optimum experience and avoid performance degradation, it is important to understand how specific Jira data dimensions influence the speed of the application. In this section we will present the results of the Jira 8.20scalability tests that investigated the relative impact of various configuration values.

### How we tested

1. As a reference for the test we used a Jira 8.20instance **with the baseline test data set** specified in "Testing methodology" and ran the full performance test cycle on it.
2. To focus on data dimensions and their effect on performance, we didn't test individual actions, but instead **used a mean of all actions** from the performance tests.
3. Next, in the baseline data set we **doubled each attribute** and ran independent performance tests for each doubled value (i.e. we ran the test with a doubled number of issues, or doubled number of custom fields) while leaving all the other attributes in the baseline data set unchanged.
4. Then, we compared the response times from the doubled data set test cycles with the reference results. With this approach we could isolate and observe how the growing size of individual Jira configuration items affects the speed of an (already large) Jira instance.

### Response times for Jira data sets

#### Jira 8.13.11 vsJira 8.20.0



**i** Response times for different data sets are similar between the Jira versions, and the differences you see on the image above aren't higher than 1%.

## Performance of Advanced Roadmaps in Jira Data Center

### Factors that affect Advanced Roadmaps performance

The biggest variable on the performance of Advanced Roadmaps in your Jira Data Center instance is simply the amount of information it contains. A plan in Advanced Roadmaps is made up of several different Jira functions, and each one must be individually executed when creating a plan. The more functions your plan executes, the longer it takes to load.

The speed of Advanced Roadmaps in your instance depends on:

- The number of issues in your instance
- The number of issue links and dependencies in your instance
- The number of issues in each plan
  - See the *Limits on plan size* section in [Plans in Advanced Roadmaps](#)
- The number of issue sources your plans have and the complexity of their JQL
  - See [Factors contributing to JQL performance in Jira server](#)
- The configuration of your hierarchy
- Any custom fields that you have enabled and use in your plans
- The number of versions and releases in your plans
  - See [Portfolio Plan is slow to interact with when loading many releases](#)

### How we tested

We used the Jira Performance Testing framework, including the hardware and methodology outlined in the [Testing methodology](#) section of this page.

Our test instance has:

- 1 million issues
- 100k users

Our test plan has:

- 500 issues
- 5 issue board issue sources
- 5 teams (1 per issue source)
- 5 level issue hierarchy
- 80 releases/versions
- 0 plan custom fields

### Summary metrics for Advanced Roadmaps in Jira Data Center 8.20

This version of Advanced Roadmaps included in Jira Data Center 8.20 produced the following load times, based on the criteria outlined above:

#### Plans using the Improved (3.x) interface:

	Advanced Roadmaps in Jira 8.20	Advanced Roadmaps in Jira 8.13
Initial load time	2983	3102
Plan backlog load time	1075	1099
Plan total load time	4063	4204
Auto-scheduling time	3594	3643

All times in ms.

#### Plans using the Legacy (2.x Live Plans) interface:

	Advanced Roadmaps in Jira 8.20	Advanced Roadmaps in Jira 8.13
Initial load time	2434	2553
Plan backlog load time	2595	2506
Plan total load time	5036	5066
Details view load time	710	702
Plan calculation time	4917	4963

All times in ms.

## Further resources

### Archiving issues

The number of issues affects Jira's performance, so you might want to archive issues that are no longer needed. You may also come to conclusion that the massive number of issues clutters the view in Jira, and therefore you still may wish to archive the outdated issues from your instance. See [Archiving projects](#).

### User Management

As your Jira user base grows you may want to take a look at the following:

- [Connecting Jira to your LDAP Directory](#) for authentication, user and group management.
- [Connecting to Crowd or Another Jira Server for User Management](#).
- [Allowing Other Applications to Connect to Jira for User Management](#).

### **Jira Knowledge Base**

For detailed guidelines on specific performance-related topics refer to the [Troubleshoot performance issues in Jira server](#) article in the Jira Knowledge Base.

### **Jira Enterprise Services**

For help with scaling Jira in your organization directly from experienced Atlassians, reach out to our [Premier Support and Technical Account Management](#) services.

### **Atlassian Experts**

The [Atlassian Experts](#) in your local area can also help you scale Jira in your own environment.

# Use a CDN with Atlassian Data Center applications

## On this page:

- [Get started with CDN](#)

### [How it works](#)

[How to determine whether a CDN will help your users](#)

[What is cached?](#)

[Planning your CDN implementation](#)

- [Infrastructure requirements](#)
- [Considerations for private instances](#)
- [Marketplace apps and third party customizations](#)

If your users are distributed across the world and experience poor performance when using Data Center products, you may be able to improve their experience by using a Content Delivery Network (CDN). Common CDNs include AWS CloudFront, Cloudflare, Akamai, and others.

CDN support is available in **Data Center** editions of:

- Jira Software 8.3
- Jira Service Management (formerly Jira Service Desk) 4.3
- Confluence 7.0
- Bitbucket 6.8.

## Get started with CDN

Here's a quick summary of what's involved to enable your CDN in Jira Data Center:

1. Use [our template](#) to spin up an AWS CloudFront distribution, or create an account with the CDN vendor of your choice.
2. Update your [load balancer and firewall](#) to allow the CDN to reach your site.
3. In your Jira Data Center application, provide the CDN URL, and enable CDN support.

As end users access your site, static assets will be cached on the edge server closest to them, and served from there until they expire. This means it might take some time before you can start measuring the impact of the CDN, depending on when your users are online and accessing the site in each location. We don't provide the ability to preload the cache, so assets will be cached as they are served for the first time.

See [Configure your CDN for Jira Data Center](#) for the full step-by-step guide.

As always, we recommend testing this on your staging environment, before making any changes to your production site.

## How it works

Static assets (such as JavaScript, CSS, and fonts) are cached on edge servers provided by a CDN vendor that are geographically closer to the user. This means when someone views a page, some of the assets needed to display the page are delivered by a server in their region, rather than from your server, known as the origin server. This can speed up page load times.



For example, if your server (known as the origin) is in Germany, a CDN can improve page load time by as much as 50% for users located in Rio de Janeiro, as static assets can be served from an edge server in Brazil. If you're new to CDNs and would like to learn more about how they work, CloudFlare provides a great introduction, see <https://www.cloudflare.com/learning/cdn/performance/>.

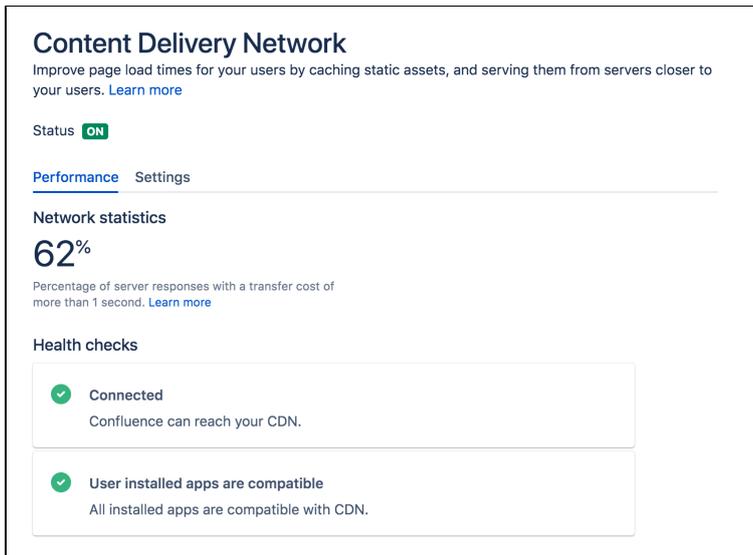
It's important to note that using a CDN will not make your application inherently faster, what it will do is reduce the load on your cluster, and reduce the latency experienced by some users, which should result in faster page load times for users.

Tests on our internal dogfooding instances located in Gdask, Poland have shown the response time for the View Issue action in Jira Data Center is ~50% faster for people accessing from US East, when CDN is enabled.

### How to determine whether a CDN will help your users

A good starting point when assessing whether a CDN will help your users, is to take a look at the network overhead experienced in your site.

Go to **Content Delivery Network** in the admin console of your Data Center application. On the **Performance** tab you'll see the percentage of requests that had a transfer cost of more than one second. Put simply, the higher the percentage, the more likely it is that your users requests are being affected by network conditions, such as latency and connection quality.



**Content Delivery Network**  
Improve page load times for your users by caching static assets, and serving them from servers closer to your users. [Learn more](#)

Status **ON**

[Performance](#) [Settings](#)

**Network statistics**  
**62%**  
Percentage of server responses with a transfer cost of more than 1 second. [Learn more](#)

**Health checks**

- Connected**  
Confluence can reach your CDN.
- User installed apps are compatible**  
All installed apps are compatible with CDN.

This network statistic is a useful indicator of the network conditions your users experience when using the product. If the percentage is high, it's likely that using a CDN will benefit your users in these conditions.

As users access pages in your site (for example a Confluence page, Jira issue, or Bitbucket pull request page), we measure the amount of time the browser has to wait to get the content of that page. We then subtract the time required to render the page on the server. This leaves us with the time it took to send the request and retrieve the response.

This time is dependent mostly on the latency between the server and the browser, but also includes things like SSL connection setup time.

This metric is collected on requests that don't use CDN, so it will continue to provide consistent statistics on your network, even after you enable CDN.

You should also consider where your users are geographically located. For example, if your servers are located in Frankfurt, and the majority of your teams are located in Germany and Austria, your team based in Malaysia may be suffering from high latency, resulting in slow page load times.

Network diagnostic tools such as `tracert`, `ping`, and `mtr` can be helpful to determine the amount of latency being experienced.

In these examples we'll use `tracert` to display some basic network statistics, including latency information. Remember to replace `yoursite.com` with your base URL.

In Windows, open Command Prompt and enter the following:

```
> tracert yoursite.com
```

In Linux or Mac OS, open Terminal and enter the following:

```
$ traceroute yoursite.com
```

This will display the number of hops, and three latency times, in milliseconds, for each server. Average the three figures to get the latency for that server.

The `mtr` command (my traceroute) is a useful combination of `ping` and `tracert`. You will need to install `mtr` to be able to use it in MacOS or Windows.

## What is cached?

We only cache static assets served by a Data Center application or Marketplace app. These are things that are only going to change when you upgrade your Data Center application or app. Dynamic content is not cached.

Here's a summary of what will be cached when you enable CDN:

Cached	Not cached
<ul style="list-style-type: none"> <li>• CSS</li> <li>• JavaScript</li> <li>• Fonts</li> </ul>	<ul style="list-style-type: none"> <li>• attached files</li> <li>• pages or issues</li> <li>• personal information, including avatars</li> <li>• assets that are part of a theme</li> </ul>

You shouldn't need to ever manually invalidate the cache, as we handle this when you upgrade your Data Center product, or an app.

## Planning your CDN implementation

### Infrastructure requirements

You can use any origin pull CDN. You're responsible for any costs associated with your CDN.

We've prepared a CloudFormation template that you can use to configure Amazon CloudFront with minimal effort. You can find all our AWS deployment resources in this repository <https://bitbucket.org/atlassian/atlassian-aws-deployment/src/master/templates/cdn/>.

There are some other infrastructure requirements that you need to be aware of before you start:

- **HTTP/2 is highly recommended**  
Your load balancer, firewall, or proxy should allow HTTP/2 traffic. Using HTTP/2 will provide the best performance for your end users. Check the documentation for your particular provider to find out how to do this.
- **Firewall considerations**  
Your CDN must be able to access and cache static assets. If your instance is not publicly accessible will you need to make some changes to your firewall to allow requests from the CDN to pass through. We recommend using application firewalls instead of standard IP range filtering, as CDN IP ranges can change without notice.

### Considerations for private instances

If your site is publicly accessible on the internet, you should be able to enable CDN without any problems.

If your site is not publicly accessible you can:

- configure your firewall to allow requests from your CDN to pass through. More information on how to do this is provided in our step-by-step guides below.
- set up your own caching servers closer to your users which will not require opening any traffic to the internet, instead of using a CDN vendor. See [How to configure Apache for caching and HTTP/2](#) to learn more about this workaround.

### Marketplace apps and third party customizations

Some marketplace apps or customizations may not be compatible with the CDN feature. A health check, on the Content Delivery Network admin screen will let you know if any of your apps are not compatible.

See [User-installed apps health check fails in Data Center when configuring CDN](#) to find out what to do if any of your apps are incompatible.

If you've developed your own plugin, see [Content Delivery Network \(CDN\) for Jira Data Center](#) for APIs you can use to confirm your plugin is compatible.

# Configure your CDN for Jira Data Center

## On this page:

- [Configure an internet facing load balancer \(optional\)](#)
  - [Add an internet-facing load balancer](#)
  - [Update your firewall rules for the internet-facing load balancer](#)
- [Configure your CDN to cache assets](#)
- [Enable CDN in Jira applications](#)
  - [Configure CDN in Jira via REST API](#)
- [Troubleshooting](#)
  - [Frequently asked questions](#)

If your users are distributed across the world and experience high latency when using Jira Software Data Center, or Jira Service Desk Data Center, you may be able to improve their experience by using a Content Delivery Network (CDN). Common CDNs include AWS CloudFront, Cloudflare, Azure CDN, Akamai, and others.

Head to [Use a CDN with Atlassian Data Center applications](#) to learn about our CDN capabilities, and how to assess whether it will improve your users' experience.

Once you're ready to start using a CDN, there are three main steps:

1. Configure an internet-facing load balancer (optional)
2. Configure your CDN.
3. Enable the CDN feature in you JIRA application.

## Configure an internet facing load balancer (optional)

If your site is not publicly accessible, you'll need to make sure that your CDN can reach it, but only to access and cache static assets. The way you do this depends on your particular load balancer and web application firewall. Refer to the documentation for your load balancer and firewall for detailed guidance.

### Add an internet-facing load balancer

Add an internet-facing load balancer to your setup. This is in addition to your primary load balancer. Your CDN is the only entity that will interact with this load balancer. We recommend you:

- Enable HTTPS - the traffic from this load balancer will be sent over the public internet and should be encrypted.
- Enable HTTP/1.1 - currently, the caching proxies and CDNs do not handle HTTP/2 well (or at all) on the way to the origin.
- For AWS deployments, you would set up an internet-facing application load balancer.

### Update your firewall rules for the internet-facing load balancer

Unlike your primary load balancer, this internet-facing load balancer must be locked down to ensure that your CDN can only pull data it is allowed to cache. When configuring your firewall rules we recommend:

1. The configuration should only allow requests for paths that start with "/s/". If your application is deployed with a context path (for example `yoursite.com/wiki` or `yoursite.com/jira`) you will need to include it in the path. All other requests must be blocked.
2. You can also choose to limit the allowed HTTP methods to GET, HEAD, OPTIONS.

For AWS deployments, you will configure a Web Access Control List (WebACL) in the Web Application Firewall attached to your application load balancer. The condition to use is a "string match condition" applied to "URI".

To check that your setup is secure, perform the following manual tests:

1. A GET on `https://internet-facing-proxy/` should return "403 FORBIDDEN".
2. A GET on `https://internet-facing-proxy/s` should return "403 FORBIDDEN".

3. A GET on `https://internet-facing-proxy/s/` should return "404 NOT FOUND".
4. A GET on `https://internet-facing-proxy/s/.` should return "403 FORBIDDEN".
5. A GET on `https://internet-facing-proxy/s/./s/` should return "404 NOT FOUND".

## Configure your CDN to cache assets

You'll need an account with a CDN provider. You're responsible for all costs associated with your CDN. We only support serving static assets from a CDN at this time. This means page content, attached files, and personally identifiable information, including things like user avatars, won't be cached by your CDN.

We've prepared a [CloudFormation template](#) that you can use to configure Amazon CloudFront with minimal effort. You can find all our AWS deployment resources in this repository <https://bitbucket.org/atlassian/atlassian-aws-deployment/src/master/templates/cdn/>.

If you choose not to use our template, define the following in your CDN configuration. This example is based on AWS CloudFront.

<b>Origin domain name</b>	This is your Atlassian application base URL, including the context path if you've configured one. For example: <code>mycompany.com/confluence</code>
<b>Origin path</b>	Leave blank. There is no need to specify a path.
<b>Allowed HTTP methods</b>	Optionally limit to: GET, HEAD, OPTIONS
<b>Viewer protocol policy</b>	redirect HTTP to HTTPS
<b>Object caching</b>	Use origin cache headers
<b>Forward cookies</b>	None This is important to make sure static assets are cached without the user context.
<b>Query String Forwarding and Caching</b>	Forward all, cache based on all
<b>HTTP protocols</b>	Must include HTTP/2
<b>Error pages/Error Caching Minimum TTL (seconds)</b>	The default error page caching time for CloudFront is 5 minutes. Consider lowering it to a value in the range of 10-30 seconds to decrease the time required to recover from an outage.
<b>Compress Objects Automatically</b>	Yes

Using the default should be fine for most of the other settings.

You will need to adapt this information for your particular CDN provider. You should refer to the documentation for your CDN for details, as we've found that terminology differs between CDNs.

## Enable CDN in Jira applications

Once you've configured your CDN, you can enable the CDN option in your Jira application.

To turn on CDN:

1. In the upper-right corner of the screen, select **Administration**  **> System**.
2. Choose **Content Delivery Network** and navigate to the **Settings** tab.
3. Set the status to **On**
4. Paste the URL generated by your CDN into the URL field and hit **Validate**.

5. If successful, save your changes.

### Content Delivery Network

Improve page load times for your users by caching static assets, and serving them from servers closer to your users. [Learn more](#)

Status **OFF**

Performance [Settings](#)

---

Status \*

On

Off

CDN URL

✔ **Connected**

Your browser can reach your CDN.

As end users access Jira, static assets will be cached on the edge server closest to them, and served from there until they expire. This means it might take some time before you can start measuring the impact of the CDN, depending on when your users are online and accessing the site in each location.

### Configure CDN in Jira via REST API

You can also interact with the CDN feature using the following REST endpoint: `<base-url>/rest/static-asset-caching/configuration`

- **GET** -returns the current CDN status, and URL.
- **DELETE** -deletes the existing configuration and reverts to the default state (CDN disabled, no URL). This is useful if you can't access the UI because of a caching problem.
- **PUT** -sets the CDN URL and status to the values passed in the body of the request as follows:

```
{
  enabled: true,
  url: https://yourcdnurl.com
}
```

### Troubleshooting

Here are some common problems that you may encounter.

- **We only accept HTTPS CDN URLs**  
This is particularly important if you're using Azure CDN, as Azure CDN will mirror the same protocol as the originating request, which means your Data Center application will need to be provisioned with HTTPS.
- **Data Center application UI is inaccessible or not functional**  
Although unlikely, a misconfiguration of your CDN or a CDN service outage may mean your application's UI is not accessible. If this happens, you will need to disable the CDN feature using the REST API, as follows.

```
curl -v -u <admin username>:<admin password> -X DELETE http://<your-base-url>/rest/static-asset-caching/configuration
```

This example uses Curl, but you can use any language. Don't forget to replace the username, password, and base URL placeholders with your own details.

- **HTTP/2 disabled**  
Your load balancer, firewall, or reverse proxy should allow HTTP/2 traffic. Using HTTP/2 will provide the

best performance for your end users. See [HTTP/2 health check fails in Data Center when configuring CDN](#) for more information.

- **User-installed apps may not be compatible**

This warning is displayed when we detect that a Marketplace or other user-installed app is using a deprecated method, which may result in assets being cached incorrectly. See [User-installed apps health check fails in Data Center when configuring CDN](#) for more information on what to do if you see this warning.

## Frequently asked questions

### Can I control what static assets are cached?

No, the application controls this. All requests for static assets are routed to the CDN. Requests for non-static assets are routed directly to your product.

### Is personally identifiable information cached?

User created content, usernames, mentions, avatars etc are not static assets, so are not cached. Your CDN should also be configured to pull content from your product with cookies stripped to make sure it operates without user context.

### Is dynamic content such as `batch.js` cached?

Although dynamically generated, `batch.js` is considered static content, so is cached.

# Jira Data Center monitoring

To help you monitor the performance of your Jira Data Center instance, we've introduced a number of monitors you can use to analyze bottlenecks and get alerts if something attention-worthy is happening.

The alerts are available in the App Diagnostics framework by going to `<JIRA_URL>/plugins/servlet/diagnostics/overview`.

- [JQL monitors](#)
- [Data Center-specific monitors: Database, Scheduler, REST resources and Servlets](#)
  - [Database monitoring](#)
  - [Scheduler monitoring](#)
  - [REST resources and Servlets monitoring](#)
- [Configuration](#)
  - [Specific configuration](#)
  - [All alerts](#)
  - [Changing default threshold](#)
  - [Enabling and disabling alerts](#)

## JQL monitors

Jira has already provided alerts for slow and complex JQL queries, but in Jira 8.4 we have added an alert for queries with a large number of custom fields. The thresholds for the below alerts are checked every time a JQL query is executed. When a slow JQL query is detected one of the below alerts are raised.

 The following monitors are available for Jira Server and Jira Data Center 8.4 and later.

Type of alert	Description	Configuration options	Alert type	Data captured
<b>Slow JQL query</b>	Default Alert raised when a JQL query is slow to execute	<code>jira.diagnostics.thresholds.slow-query-millis</code> <ul style="list-style-type: none"> <li>• Threshold in milliseconds defined for slow JQL query alerts</li> <li>• Default <b>5000</b></li> </ul>	Info	<ul style="list-style-type: none"> <li>• Thread dump</li> <li>• Number of clauses per field</li> <li>• Number of clauses in query</li> <li>• Number of custom fields</li> <li>• Number of results</li> <li>• Execution time in milliseconds</li> </ul>

<p><b>Complex JQL query</b></p>	<p>Raised when a complex JQL query is executed.</p>	<p><code>jira.diagnostics.thresholds.query-complexity</code></p> <ul style="list-style-type: none"> <li>• Query complexity alert threshold defined in the number of Lucene query clauses.</li> <li>• Default <b>10000</b></li> </ul>	<p>Info</p>	<ul style="list-style-type: none"> <li>• Thread dump</li> <li>• Number of clauses per field</li> <li>• Number of clauses in query</li> <li>• Number of custom fields</li> <li>• Number of results</li> <li>• Execution time in milliseconds</li> </ul>
<p><b>Large number of request</b></p>	<p>Raised when a JQL query produces a large number of results.</p>	<p><code>jira.diagnostics.thresholds.number-of-results</code></p> <ul style="list-style-type: none"> <li>• Threshold in the number of returned issues from a JQL query for JQL alerts</li> <li>• Default <b>1000</b></li> </ul>	<p>Info</p>	<ul style="list-style-type: none"> <li>• Thread dump</li> <li>• Number of clauses per field</li> <li>• Number of clauses in query</li> <li>• Number of custom fields</li> <li>• Number of results</li> <li>• Execution time in milliseconds</li> </ul>

<b>Large number of custom fields</b>	Raised when a JQL query contains a large number of custom fields.	<p><code>jira.diagnostics.thresholds.query-custom-fields</code></p> <ul style="list-style-type: none"> <li>• Threshold for the number of custom fields used in a JQL query</li> <li>• Default <b>1000</b></li> </ul>	Info	<ul style="list-style-type: none"> <li>• Thread dump</li> <li>• Number of clauses per field</li> <li>• Number of clauses in query</li> <li>• Number of custom fields</li> <li>• Number of results</li> <li>• Execution time in milliseconds</li> </ul>
--------------------------------------	---	--	------	--

### Data Center-specific monitors: Database, Scheduler, REST resources and Servlets

 The following monitors have been introduced in Jira 8.4 and are available for Jira Data Center 8.4 and later.

#### Database monitoring

The following are the alerts for slow database queries, high pool utilization, and abandoned connections:

Type of alert	Description	Default Threshold	Configuration options	Alert type	Data captured
<b>Connection leak</b>	Raised when a database connection has leaked from the pool.	300 seconds	Configurable via the <code>tomcat pool-remove-abandoned-timeout</code> option.	Warning	<ul style="list-style-type: none"> <li>• Active connection count</li> <li>• Idle connection count</li> <li>• Max pool size</li> </ul>

<p><b>High connection pool utilization alert</b></p>	<p>Raised when the database connection pool is utilized at the limit for a specified period of time.</p> <p>Has a default polling interval of 5 seconds</p>	<p>90% utilization for 15 minutes</p>	<p>jira.diagnostics.threshold.database-pool-utilization</p> <ul style="list-style-type: none"> <li>• Percentage threshold for raising an alert on database pool utilization.</li> <li>• Default<b>80</b></li> </ul> <p>jira.diagnostics.threshold.database-pool-utilization-time-window</p> <ul style="list-style-type: none"> <li>• Time window in minutes for tracking database pool utilization</li> <li>• Default<b>15 minutes</b></li> </ul> <p>jira.diagnostics.settings.database-pool-poller-interval</p> <ul style="list-style-type: none"> <li>• Poller interval for database pool in seconds</li> <li>• Default<b>5</b></li> </ul>	<p>Info</p>	<ul style="list-style-type: none"> <li>• Active connection count</li> <li>• Idle connection count</li> <li>• Max pool size</li> </ul>
--	---	---------------------------------------	--	-------------	---

<b>Slow operation alert</b>	Raised when an SQL operation is slow to execute. Optionally, the SQL statement being executed can also be captured but this is configurable and is disabled by default.	5 seconds with SQL not included in alerts	<p>jira.diagnostics.settings.include-sql-in-alerts</p> <ul style="list-style-type: none"> <li>Settings to include potentially exposed SQL queries in diagnostic alerts</li> <li>Default <b>false</b></li> </ul> <p>jira.diagnostics.thresholds.slow-query-millis</p> <ul style="list-style-type: none"> <li>Threshold in milliseconds defined for slow JQL query alerts</li> <li>Default <b>5000</b></li> </ul>	Info	<ul style="list-style-type: none"> <li>SQL statement (optional)</li> <li>Execution time in milliseconds</li> <li>Plugins involved</li> <li>Thread Name</li> </ul> <p>Note: The data such as URL or username are not currently captured.</p>
-----------------------------	---	---	---	------	---

### Scheduler monitoring

The following are the alerts for high pool utilization and slow jobs:

Type of alert	Description	Default Threshold	Configuration options	Alert type	Data captured
---------------	-------------	-------------------	-----------------------	------------	---------------

<p><b>High utilization alert</b></p>	<p>Raised when the scheduler worker threads are 100% utilized for a period of time.</p> <p>Has a default polling interval of 5 seconds.</p>	<p>More than or equal to 100% utilization for 15 minutes</p>	<p><code>jira.diagnostics.threshold.scheduler-utilization-time-window</code></p> <ul style="list-style-type: none"> <li>• Time window in minutes for tracking high scheduler utilization</li> <li>• Default <b>15</b></li> </ul> <p><code>jira.diagnostics.settings.scheduler-poller-interval</code></p> <ul style="list-style-type: none"> <li>• Poller interval for scheduler diagnostics in seconds</li> <li>• Default <b>5</b></li> </ul>	<p>Info</p>	<ul style="list-style-type: none"> <li>• Number of worker threads</li> <li>• Details of all executing jobs including: <ul style="list-style-type: none"> <li>• Job runner key</li> <li>• Job ID</li> <li>• Running time</li> <li>• Job Runner class (if available)</li> <li>• Plugin of origin (if available)</li> </ul> </li> </ul>
--------------------------------------	---	--	---	-------------	--

<b>Slow job alert</b>	<p>Raised when a scheduled job is taking longer than its configured interval to execute.</p> <p>Has a default polling interval of 5 seconds.</p>		<p><code>jira.diagnostics.settings.scheduler-poller-interval</code></p> <ul style="list-style-type: none"> <li>• Poller interval for scheduler diagnostics in seconds</li> <li>• Default<b>5</b></li> </ul>	Info	<ul style="list-style-type: none"> <li>• Job runner key</li> <li>• Job ID</li> <li>• Execution time</li> <li>• Job Runner class (if available)</li> <li>• Plugin of origin (if available)</li> </ul>
-----------------------	--	--	---	------	--

### REST resources and Servlets monitoring

The following is the alerts for slow running http requests:

Type of alert	Description	Default Threshold	Configuration options	Alert type	Data captured
<b>Slow request</b>	Raised when a HTTP request is slow to respond.	60 seconds	<p><code>jira.diagnostics.settings.http-request-max-elapsed-time</code></p> <ul style="list-style-type: none"> <li>• Threshold for in seconds for long running HTTP requests</li> <li>• Default<b>60</b></li> </ul>	Info	<ul style="list-style-type: none"> <li>• Request path</li> <li>• Username</li> <li>• Execution time in milliseconds</li> <li>• Thread ID</li> <li>• Thread status</li> </ul>

## Configuration

### Specific configuration

Unless described in the table, each of the above mentioned alerts is configurable through the `jira-config.properties` file located in the `<jira-home>` directory. For more information on how to configure the file, see [jira-config.properties file](#).

### All alerts

Apart from the specific configuration described for each of the available alerts, you can change the general alert configuration. This way, your changes will apply to all alerts. To do that, open `jira-config.properties` and edit the following metrics:

-com.atlassian.jira.health.diagnostics.alerts.retention-period-days: number of days to retain Diagnostics Alerts; 30 by default  
 - com.atlassian.jira.health.diagnostics.alerts.truncation-interval-days: number of days to run the truncation job; 1 by default

For more information on how to configure the file, see [jira-config.properties file](#).

### Changing default threshold

To change the default threshold, you need to edit the following properties in the [jira-config.properties file](#):

- jira.diagnostics
- com.atlassian.jira.health.diagnostics.

### Enabling and disabling alerts

The monitors are enabled by default, but they can be disabled and enabled with FeatureFlags. Learn more about how to work with [dark features](#).

Feature type	Feature name
All monitors	com.atlassian.diagnostics.monitors
JQL	com.atlassian.diagnostics.jql.monitor
Database	com.atlassian.diagnostics.database.monitor
Scheduler	com.atlassian.diagnostics.scheduler.monitor
REST resources and Servlets	com.atlassian.diagnostics.http.monitor

# Security overview and advisories

This document is for system administrators who want to evaluate the security of the Jira application. The page addresses overall application security and lists the security advisories issued for Jira. As a public-facing web application, Jiras application-level security is important. This document answers a number of questions that commonly arise when customers ask us about the security of our product.

Other topics that you may be looking for:

- For information about permissions in Jira, see [Permissions overview](#).
- For guidelines on configuring the security of your Jira site, see [Server optimization](#).

## Skip to

- [Application Security Overview](#)
- [Finding and Reporting a Security Vulnerability](#)
- [Publication of Jira Security Advisories](#)
- [Severity Levels](#)
- [Our Security Bugfix Policy](#)
- [Security Advisories](#)

## Application Security Overview

### Password Storage

When Jiras internal user management is used, passwords are hashed through the salted [PKCS5S2 implementation provided by Embedded Crowd](#) before being stored in the database. There is no mechanism within Jira to retrieve a user's password when password recovery is performed, a reset password link is generated and mailed to the user's registered address.

When external user management is enabled, password storage is delegated to the external system.

### Buffer Overflows

Jira is a 100% pure Java application with no native components. As such it is highly resistant to buffer overflow vulnerabilities possible buffer overruns are limited to those that are bugs in the Java Runtime Environment itself.

### SQL Injection

Database queries are generated using standard APIs for parameter replacement rather than string concatenation. As such, Jira is highly resistant to SQL injection attacks.

### Script Injection

Jira is a self-contained Java application and does not launch external processes. As such, it is highly resistant to script injection attacks.

### Transport Layer Security

Jira does not directly support SSL/TLS. Administrators who are concerned about transport-layer security should set up SSL/TLS at the level of the Java web application server, or the HTTP proxy in front of the Jira application.

For more information on configuring Jira for SSL, see [Running Jira over SSL or HTTPS](#).

### Session Management

Jira delegates session management to the Java application server in which it is deployed. We are not aware of any viable session-hijacking attacks against the Tomcat application server shipped with Jira.

## Apps (add-ons) Security

Administrators install third party apps **at their own risk**. Apps run in the same virtual machine as the Jira server, and have access to the Java runtime environment, and the Jira server API.

Administrators should always be aware of the source of the apps they are installing, and whether they trust those apps.

## Administrator Trust Model

Jira is written under the assumption that anyone given [System Administrator privileges](#) is trusted. System administrators are able, either directly or by installing plugins, to perform any operation that the Jira application is capable of.

As a security best practice, you should not run Jira as the root/Administrator user. If you want Jira to listen on a privileged network port, you should set up port forwarding or proxying rather than run Jira with additional privileges. The extra-careful may consider running Jira in a virtualized environment.

## Stack Traces

To help when debugging a problem, Jira provides stack traces through the web interface when an error occurs. These stack traces include information about what Jira was doing at the time, and some information about your deployment server.

Only non-personal information is supplied such as operating system and version and Java version. With proper network security, this is not enough information to be considered dangerous. No usernames or passwords are included.

## Finding and Reporting a Security Vulnerability

Atlassian's approach to reporting security vulnerabilities is detailed in [How to Report a Security Issue](#).

## Publication of Jira Security Advisories

Atlassian's approach to releasing security advisories is detailed in [Security Advisory Publishing Policy](#).

## Severity Levels

Atlassian's approach to ranking security issues is detailed in [Severity Levels for Security Issues](#).

## Our Security Bugfix Policy

Our approach to releasing patches for security issues is detailed in our [Security Bugfix Policy](#).

## Security Advisories

There are no new security advisories for Jira. To see all Atlassian security advisories, go to [Security Advisories](#).

# Getting help

## How can we help you?

We have a number of [help resources](#) available. You can get your problem resolved faster by using the appropriate resource(s).

[I don't know how to do something](#)

[Something isn't working](#)

[I don't like the way something works](#)

[Something else?](#)

### I don't know how to do something

1. Search [Atlassian Answers](#).
2. Raise a [support request](#)\*

### Something isn't working

1. Check the Jira knowledge base at <https://confluence.atlassian.com/display/JIRAKB>.
2. Search [Atlassian Answers](#).
3. Raise a [support request](#)\*

If you've identified a bug but don't need further assistance, raise a [bug report](#).

### I don't like the way something works

Raise a [suggestion](#).

### Something else?

If you need help with something else, raise a [support request](#)\*

*\* If you are the **Jira system administrator**, you have a number of additional support tools available. See [Raising support requests as an administrator](#) for details.*

## About our help resources

### Atlassian Support

Our support team handles support requests that are raised in our [support system](#). You need to log in using your [Atlassian account](#) before you can raise support requests.

For information on our general support policies, including support availability, SLAs, bugfixes, and more, see [Atlassian Support Offerings](#). Note, you'll find anything security-related at [Security@ Atlassian](#).

### Atlassian Answers

[Atlassian Answers](#) is our official application forum. Atlassian staff and Atlassian users contribute questions and answers to this site.

You may be able to find an answer immediately on Atlassian Answers, instead of having to raise a support request. This is also your best avenue for help if:

- you are using an unsupported Jira instance or an unsupported Jira platform,
- you are trying to perform an unsupported operation, or
- you are developing an app for Jira Software.

You can also have a look at the [most popular Jira answers](#).

### Jira knowledge base

If there are known issues with a version after it has been released, the problems will be documented as articles in our [knowledge base](#).

### **Atlassian issue tracker**

Our official [issue tracker](#) records our backlog of bugs, suggestions, and other changes. This is open for the public to see. If you log in with your Atlassian account, you will be able to create issues, comment on issues, vote on issues, watch issues, and more.

Tip: Before you create an issue, search the existing issues to see if a similar issue has already been created.