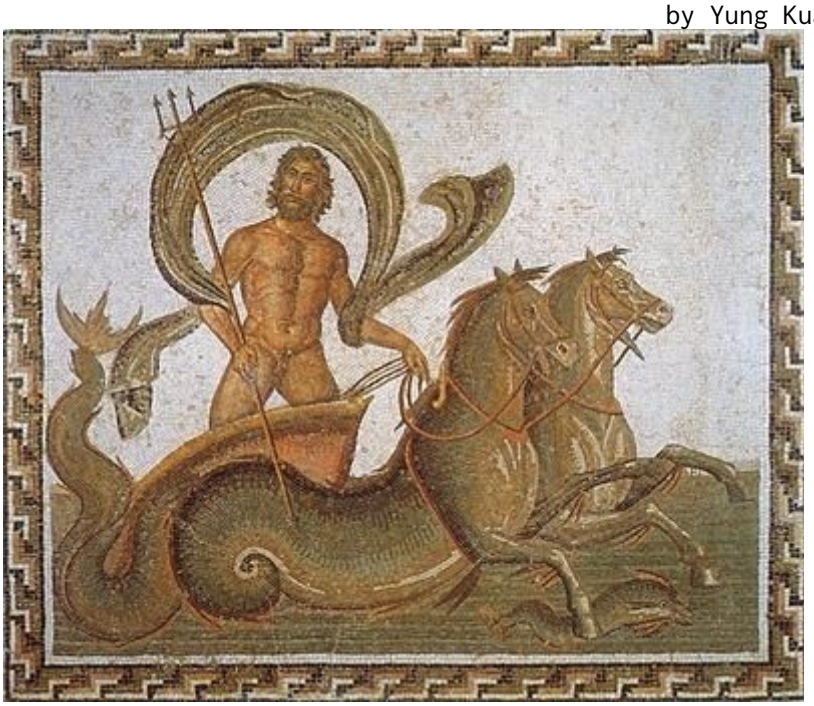
poseidon manual

by Yung Kuan Hsu 2020,2021



What is poseidon?

Poseidon is an D language IDE start from 2005 by Shawn Liu on Dsource (http://www.dsource.org/projects/poseidon), I join the project since 2006, poseidon code by Digitalmars D V1.0 and DWT(SWT port to D)

By many reasons, I abort the project since 2011(and Dsource is getting down…), anyway, I think I've done and reach my target.(I think use a program language to build an IDE, then use the IDE coding the IDE sources itself is a cool stuff!)

About 2015, I think coding on linux and cross-platform is also a cool stuff, I tried to test coding on linux, I thought of the previous concern about the freeBASIC compiler and the first contact with the BASIC language since I was a child, at that time fbedit had stopped developing, I came to try to develop a freeBASIC IDE, the usable poseidonD was developed first, and then it was poseidonFB!

Support:

1. Syntax highlighting
2. Project manager
3. Autocompletion & Calltip
4. Function / Type / Variable... treeview
5. Find / Replace in document or project
6. Jump to defintion
7. Utf-8/16/32 decode & encode
8. Debug
9. Compile / Quick run / Build project
10. etc.....

Environment:

Windows:
Digital Mars D V1.076 32bit with Tango standard library
IUP 3.27 with Modified iup_scintilla.dll
Windows 7 64bit

Linux:
Digital Mars D V1.074 64bit with Tango standard library
IUP 3.27(GTK2 / GTK3) with Modified iup_scintilla.so
Linux mint 17~19.3 64bit
gcc 4.8 (D backend)
Ubuntu 12.04 64bit and gcc 4.6 (Appimage GTK2)
Ubuntu 16.04 64bit and gcc 4.8 (Appimage GTK3)

poseidonFB:
https://bitbucket.org/KuanHsu/poseidonfb/

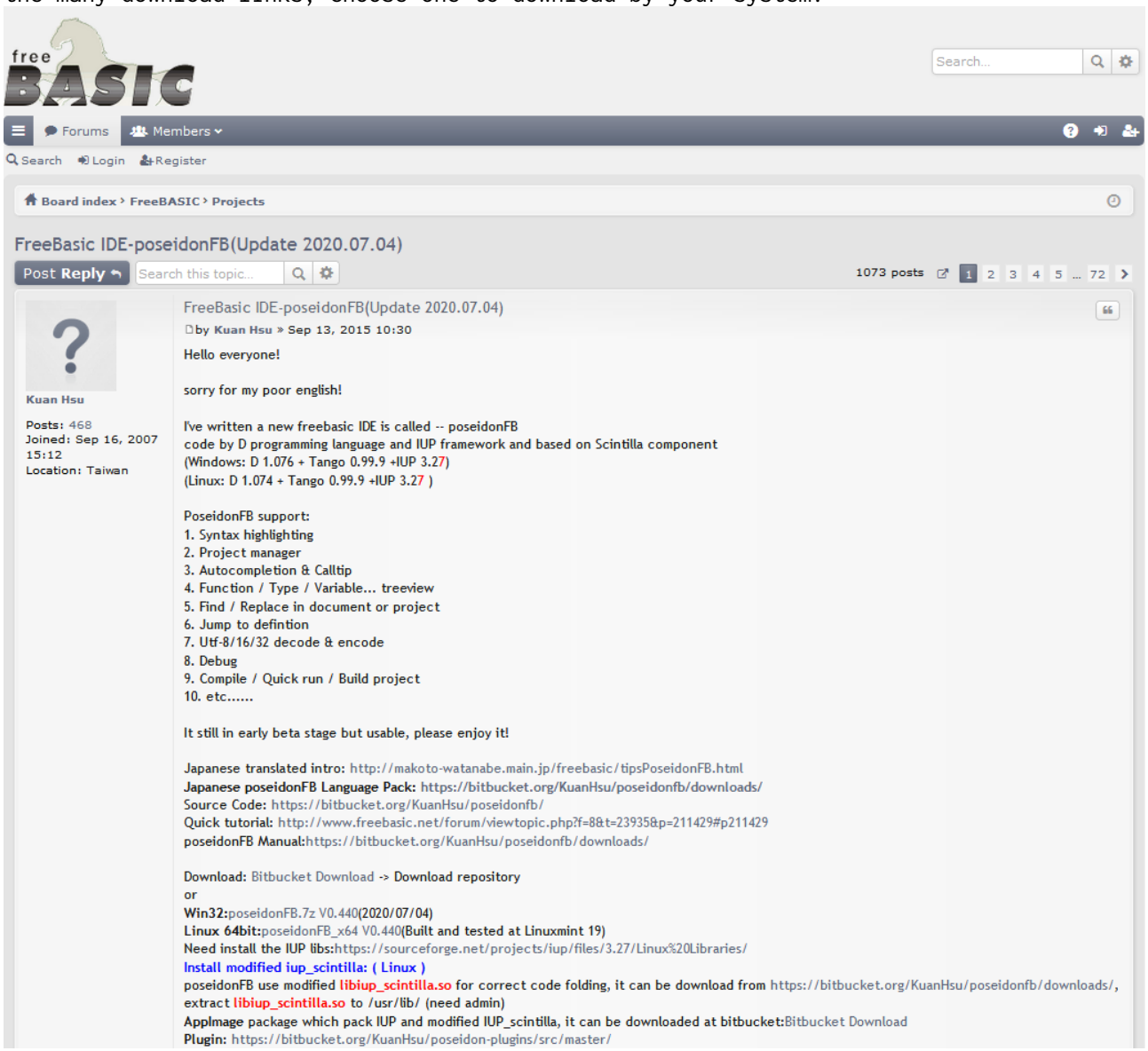
poseidonD:
https://bitbucket.org/KuanHsu/poseidond/

Kuan Hsu's mail:
nagahiro.kyo@gmail.com

freeBASIC forums:
https://www.freebasic.net/forum/viewtopic.php?f=8&t=23935

Install

Go to https://www.freebasic.net/forum/viewtopic.php?f=8&t=23935, the first post show the many download links, choose one to download by your system.



Windows:

On Windows, poseidon built under 32 bits D compiler, we can get poseidon(FB/D).7z(mediafire) or poseidon(FB/D)_revXXX.7z(bitbucket), just extract it to our folder, run the poseidon(FB/D).exe

The archive include iup_scintilla.dll, it's a modified version for folding and comment and only support(freebasic/d lerxer)

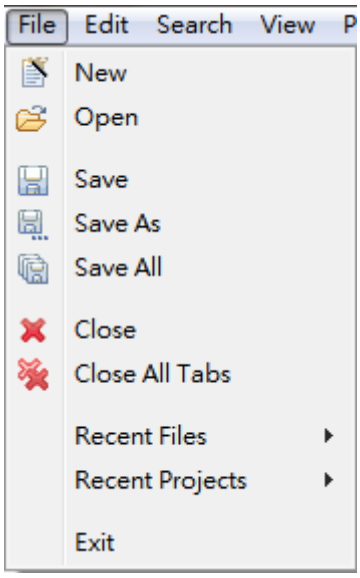
Linux:

The easier way is download the appimage, it pack the GTK2 / GTK3 and modified IUP libraries so no need to install IUP libraries, after downloading, set the *.appimage Permission **"Allow executing file as program"** is ON, please see https://discourse.appimage.org/t/how-to-run-an-appimage/80

The second way is download the IUP and extract to folder, download the modified iup_scintilla.so to the folder and replace old one, open terminal key in(Ubuntu): sudo ./install to install IUP, then download poseidon(FB/D)_x64.tar.gz or poseidon(FB/D)_x64_revXXX.tar.gz and extract it to our another folder, run poseidon(FB/D)_x64.

Menu

File



New:
Create a new document.(named in NONAME#n.bas)

Open:
Open a existed file.

Save:
Save file.

Save As:
Save and rename file.

Save All:
Save all documents.(All in MainTab & SubTab)

Close:
Close active(focus) file.

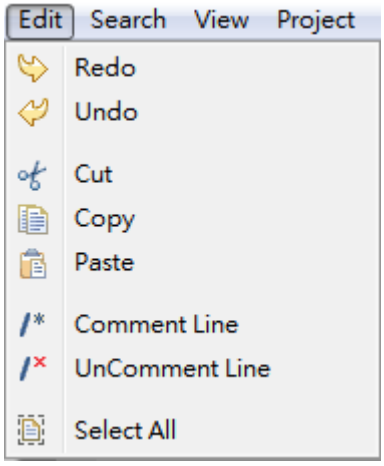
Close All Tabs:
Close all documents.(All in MainTab & SubTab)

Recent Files:
Quick open file.

Recent Projects:
Quick open project.

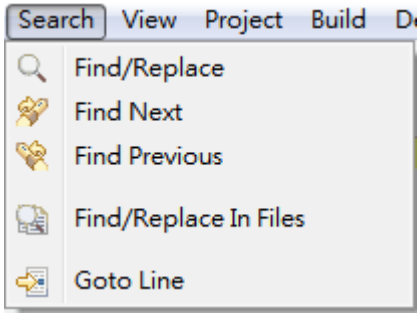
Exit:
Bye, poseidon~

Edit

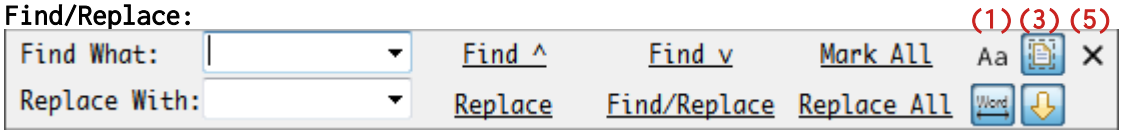


I think no need to explain...

Search



Find/Replace:



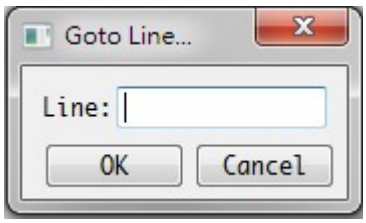
A dialog show at Bottom of document.

Find ^: Find previous
Find v: Find next
Mark All: Bookmark All
Replace: Replace the selected text, if no selection, it does nothing.
Find/Replace: Find the word and replace it.
Replace All: Force replace all fit word in document.

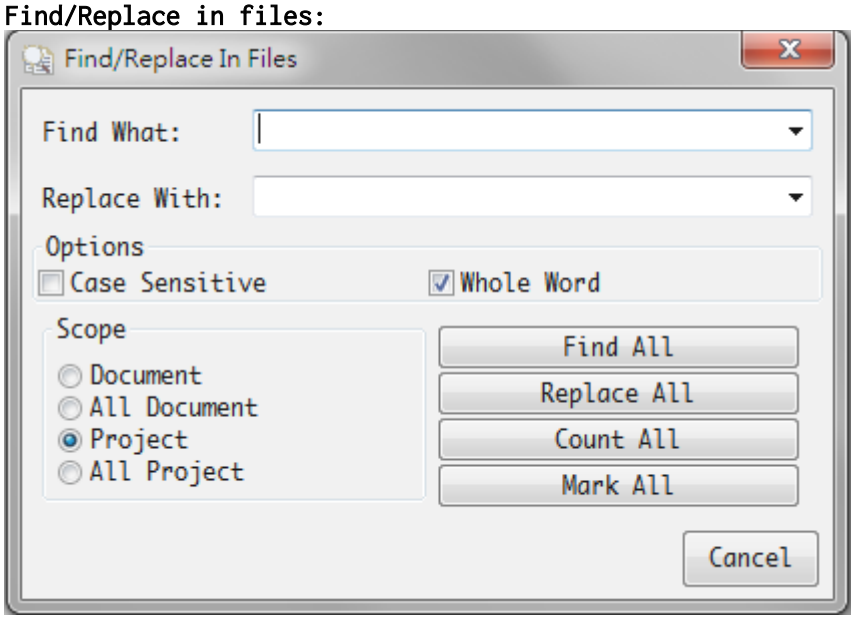
(1)Case Sensitive ON / OFF
(2)Whole Word ON / OFF
(3)Replace Scope, if the button is pressed, the scope is all document or in selection.
(4)Find/Replace direction
(5)Hide

Find Next
Find Previous:
By searching selected word.

Goto Line:



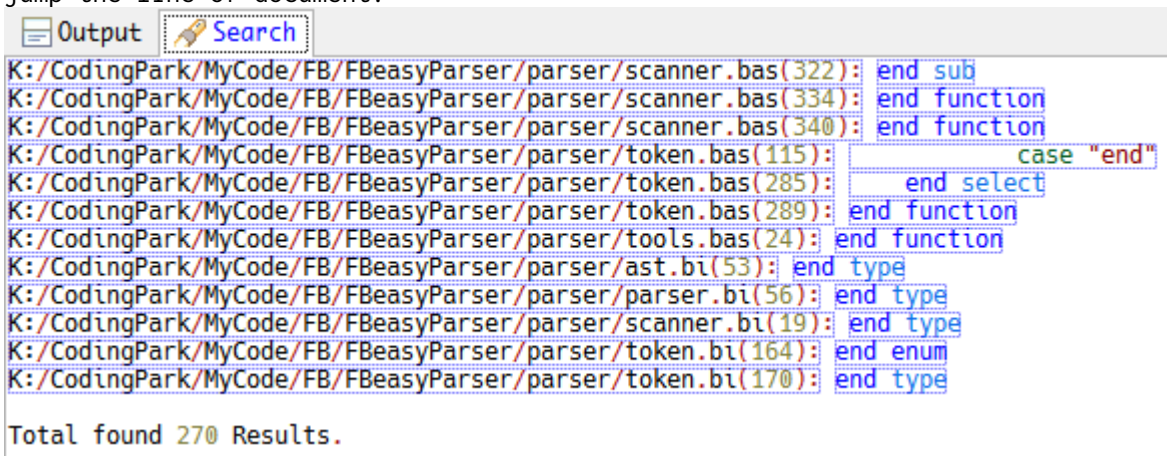
Find/Replace in files:



Options:
Case Sensitive
Whole Word

Scope:
Document: Active(focus) document.
All Document: All opened documents.
Project: Active(focus) project.
Project: A;; loaded projects.

Find All:
The Search panel will show the results, we can double-click the list item on panel to jump the line of document.

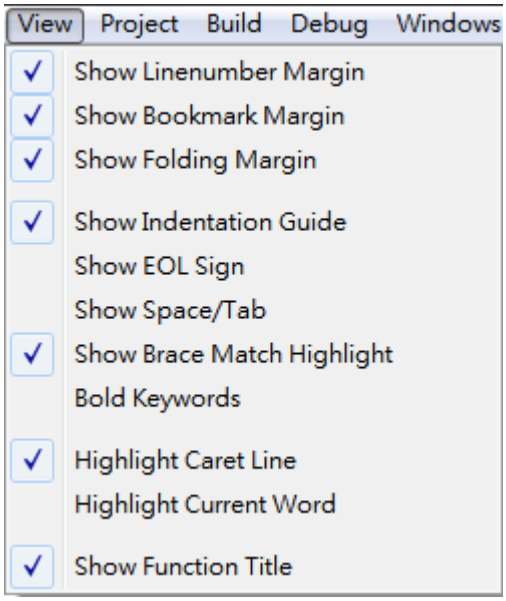


Replace All:
Notice that it can’t be undo.

Count All:
Just show the count result.

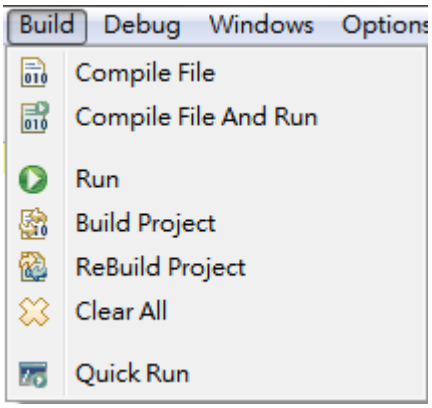
Mark All:
Bookmarks, it works on Scope= Document / All Document.

View



This items is quick selector of Preference.

Build



Compile File:
Compile a single file, if the document is NONAME#n.bas, poseidon will ask us save first.

Compile File And Run:
After Compiling, run the execute.

Run:
Run the execute.

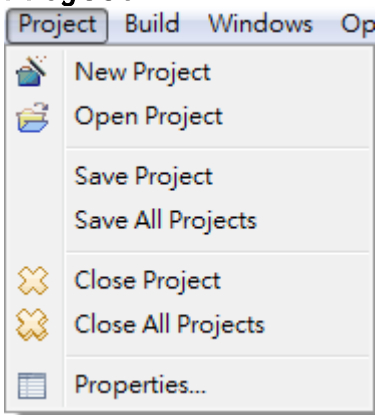
Build Project:
Build the project, if object file existed, linked with it without compile.

ReBuild Project:
Build the project, all sources will be re-compiled.

Clear All:
Clear object and exeute files

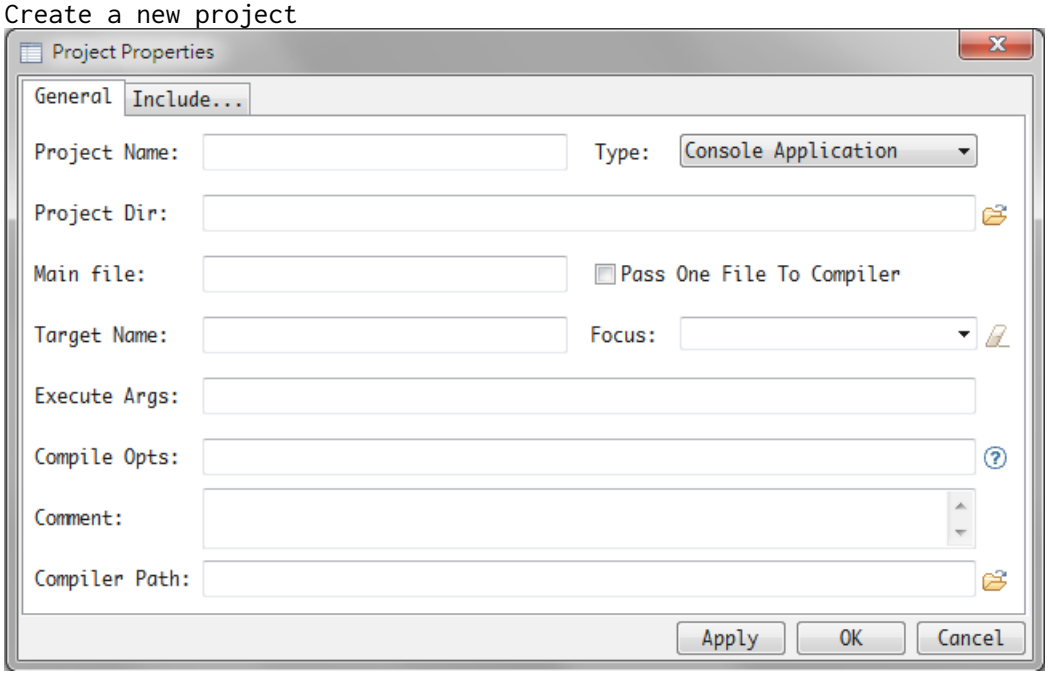
Quick Run:
Compile a single file and run it, the tempfile will be deleted after running.

Project



poseidon's project is folder-base, all project information is put in one FB.poseidon / D.poseidon file, so one folder is one project.

New Project:
Create a new project



Project Name:
The Name of project.

Type:
Console Application / Static library / Dynamic link library.

Project Dir:
Just setting once while create new project, then the project working path is where the FB.poseidon / D.poseidon is.

Main File:
Set main file without extension, the entry point.

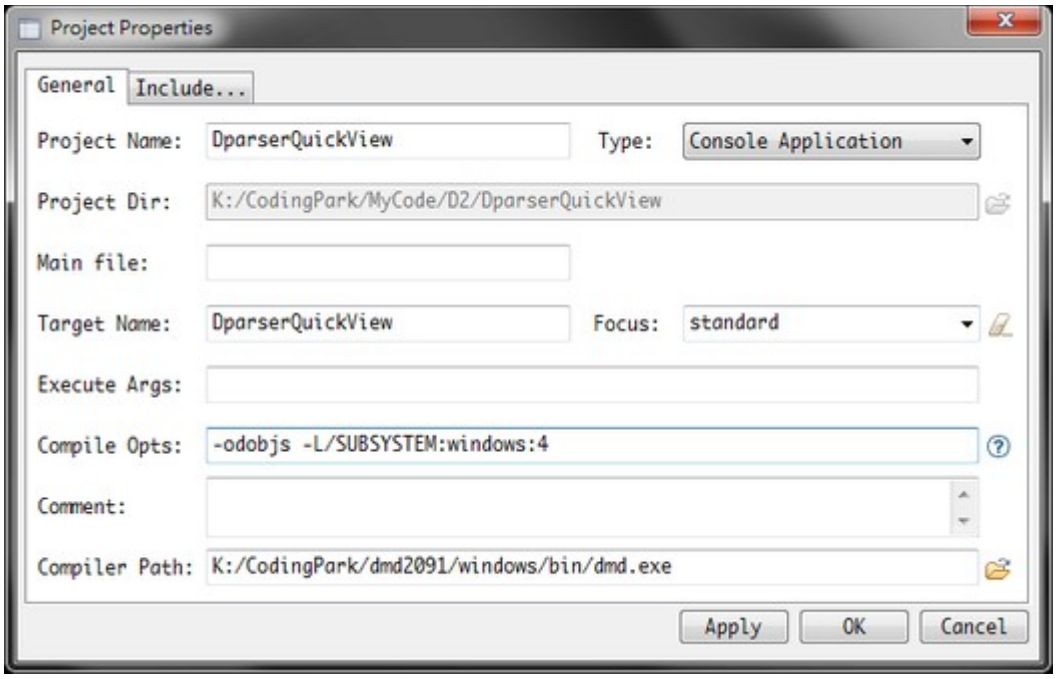
Pass One File To Compiler:
If checked, poseidon will pass one file to compiler, need set main file first.

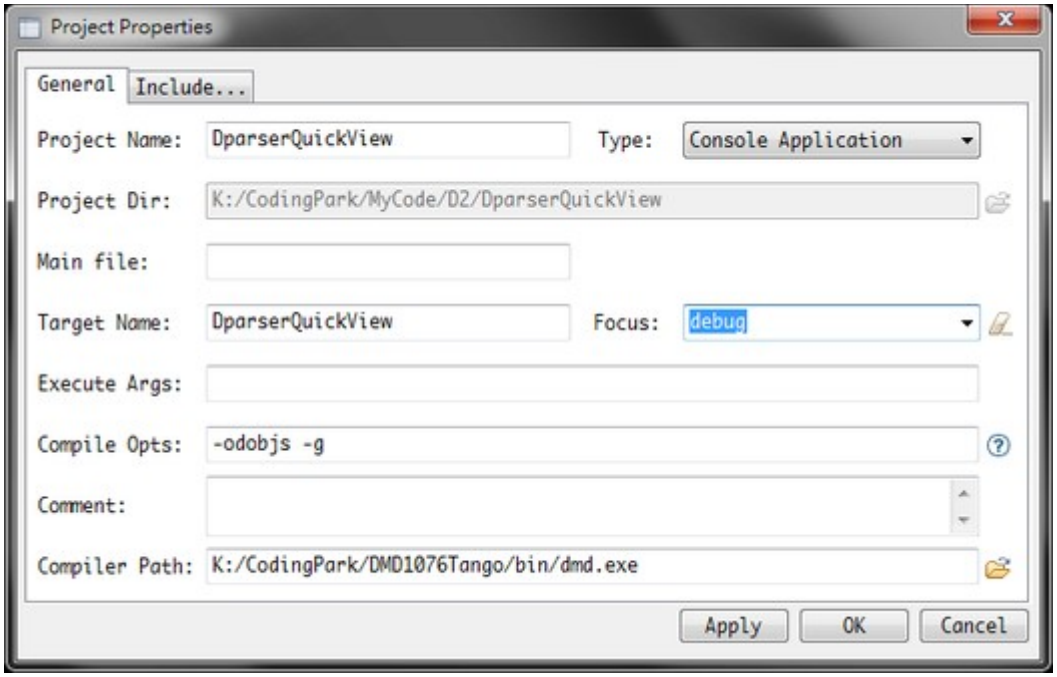
Target Name:
poseidonFB will pass -X to compiler, poseidonD will pass -ofTargetName to compiler, no need to set the suffix, without target name, poseidon set the project name default.

Compile Opts:
Set the compiler options, eq: -s gui (poseidonFB) / -odobjs -L/SUBSYSTEM:windows(poseidonD)

Compiler Path:
Set the compiler fullpath of this project, poseidonFB will set ../inc(Windows) or ../freebasic/include(linux) is include file path, poseidonD will search sc.ini(Windows) or sc.conf(linux) to load default modules, like phobos or tango.

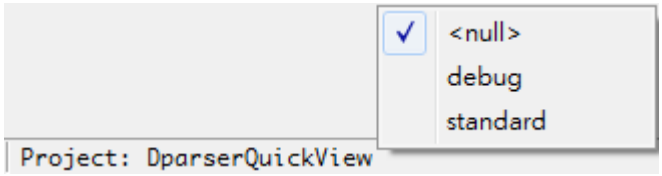
Focus:
Muliple settings of Compile Opts / Compiler Path / Include Paths / Libraries paths



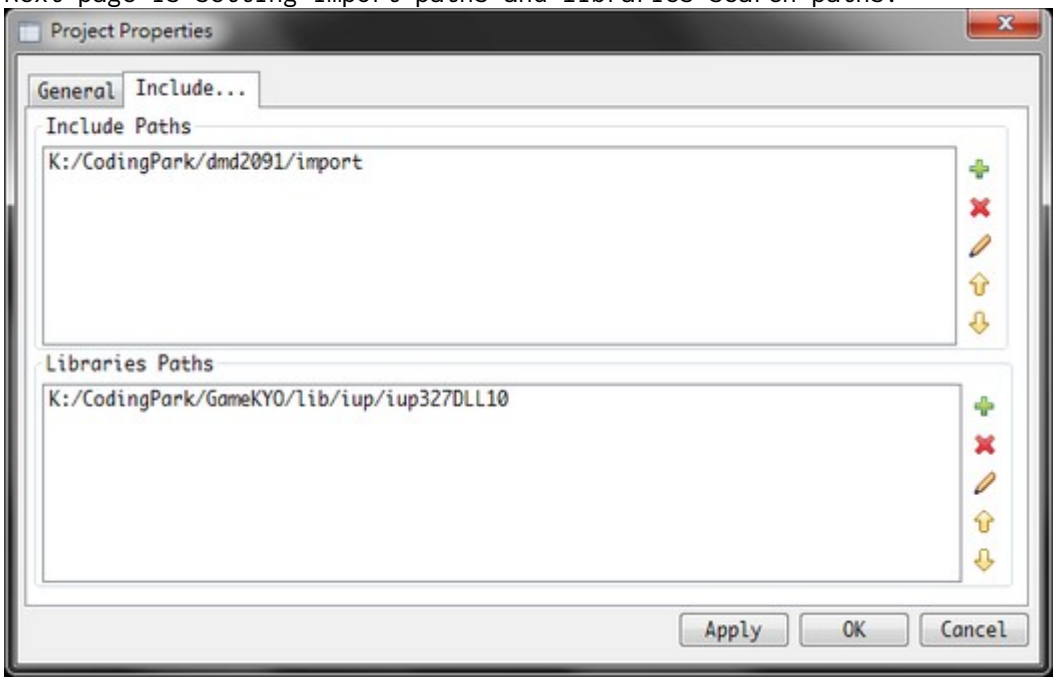


Set the name of “Focus” and Compile Opts / Compiler Path, then click “Apply” to save, use dropdown button to select, eraser button to delete.

We can also quick select the “Focus” without enter the Project Properties dialog, in the bottom of poseidonD, right-click the project name to quick select.



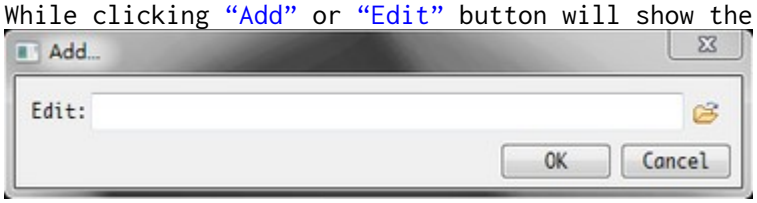
Next page is setting import paths and libraries search paths.



Include Paths:
Pass -i<name> to compiler(poseidonFB) / -Ipath to compiler(poseidonD).

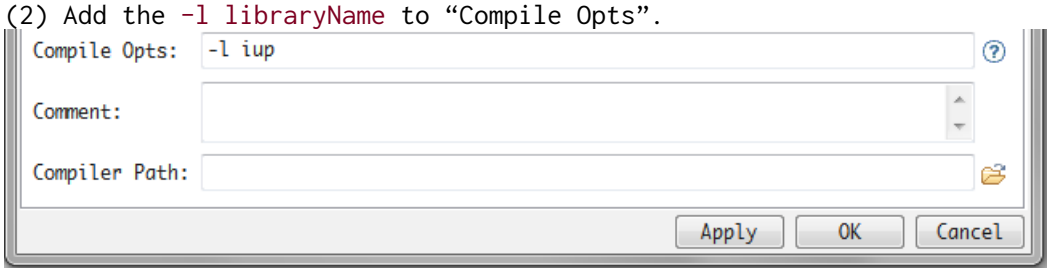
Libraries Paths:
Pass -p<name> to compiler(poseidonFB) / -L-Lpath to compilers(poseidonD)

Click “Add” to add path to list.
Click “Del” to del path from list.
Click “Edit” to edit existed path from list.
Click “Arrow” to change path sort in list.
While clicking “Add” or “Edit” button will show the sub dialog:

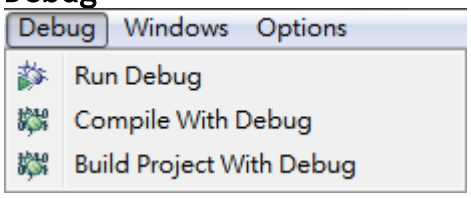


We can directly set the path or click the “Open” button to select the path.

If we want to add libraries to link, there are two ways in poseidonFB:
(1) Add #inclib "libname"
(2) Add the -l libraryName to “Compile Opts”.



Debug

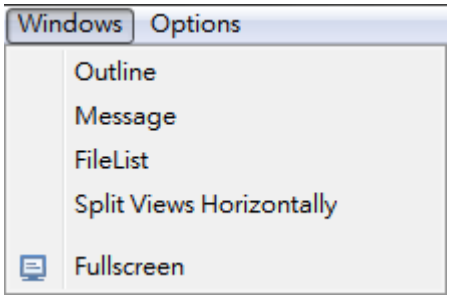


poseidonFB use GDB to do debug.

Run Debug:
gdb exefile command.

Compile With Debug / Build Project With Debug:
Compile with -g option.

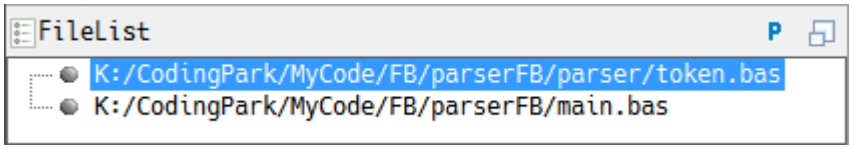
Windows:



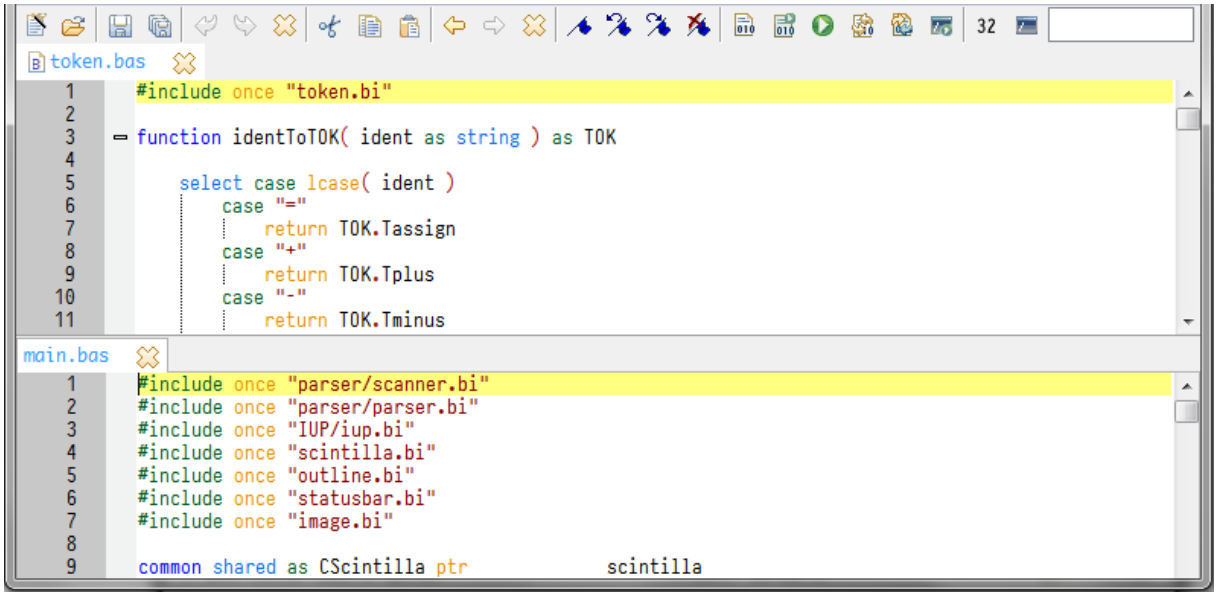
Outline:
Show / hide left panel(Project and Outline)

Message:
Show / hide bottom panel(Output and Search), we can also double-click the status bar.

FileList:
Show / hide file list.



Split Views Horizontally:
poseidon support 2 tabs, set this option be checked, the arrangement be changed.

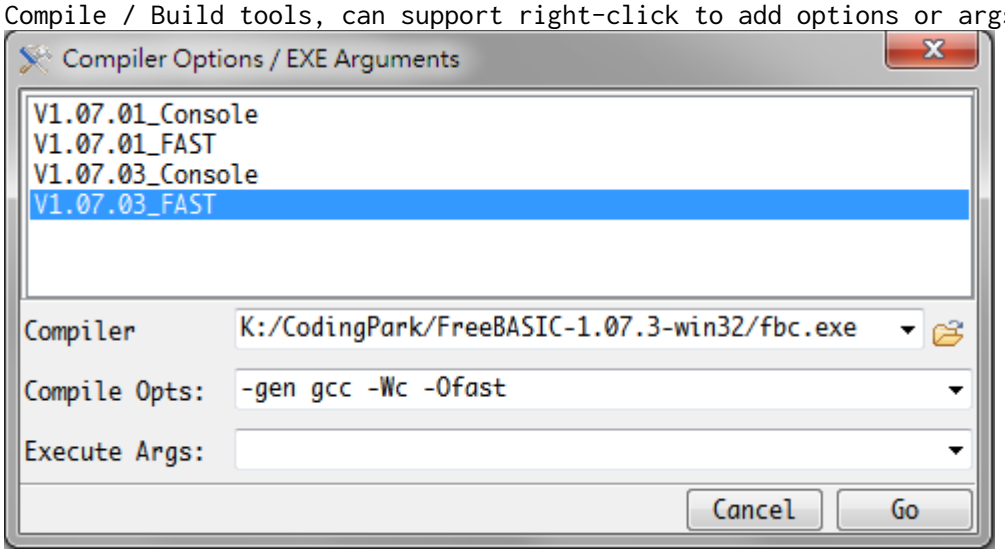


Toolbar

Blue Area:
Navigation Cache, can backward / forward / clear.

Green Area:
Bookmark, Add / Backward Find / Forward Find / Clear.

Read Area:
Compile / Build tools, can support right-click to add options or args quickly.



The list can be set at **“Custom Compiler Options”**(page 20), the compiler / opts / args can be quick set by click the items.

Also we can key-in the compiler / opts / args by ourselves, the value can be record and use dropdown button do quick select.

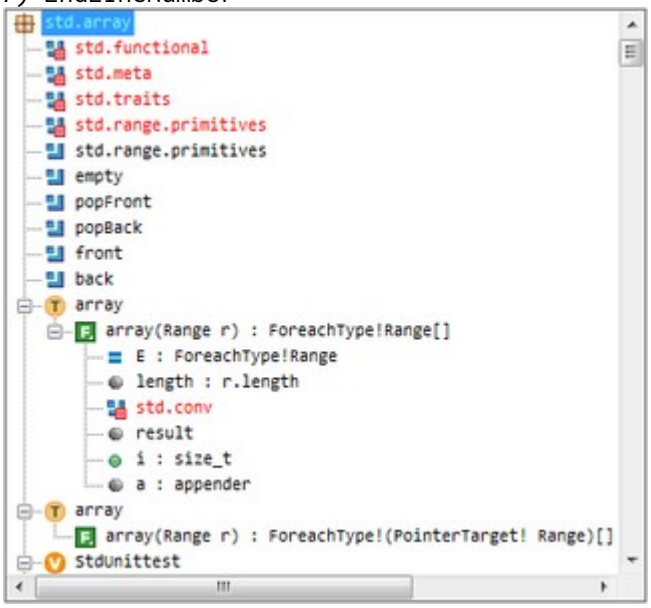
(1) 32 / 64 bit
poseidon will use 32 / 64 bits compiler.

(2) Console / GUI
poseidon will pass -s console / -s gui option to compiler.

Parser

poseidon with built-in tokens scanner and parser. At first loading and scan the file to produce tokens, then parse tokens the produce the parsedTree directly not the AST(Abstract Syntax Tree).

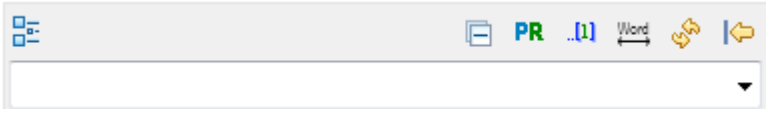
Every parsedTree node with below information:
(1) Kind
(2) Name
(3) Protection
(4) Type (with Parameter)
(5) Base
(6) LineNumber
(7) EndLineNumber



We can read the node information of tree to implement the codecompletion and calltip and jump the definite and show the type information etc. The struct of parsedTree is shown as poseidonFB/D outline:
The first node is D_MODULE kind node, next is it's children layer are many D_IMPORT and D_TEMPLATE(Aggregate Templates with child D_FUNCTION)x2 and D_VERSION, the child of D_TEMPLATE is D_FUNCTION with it's children layer D_ALIAS, D_VARIABLE…

If we double-click the node will read the LineNumber information, the scroll the document to specific line. (There is a pointer to parsedTree node in IUP treenode)

The "Outline" panel with a toolbar / search list:

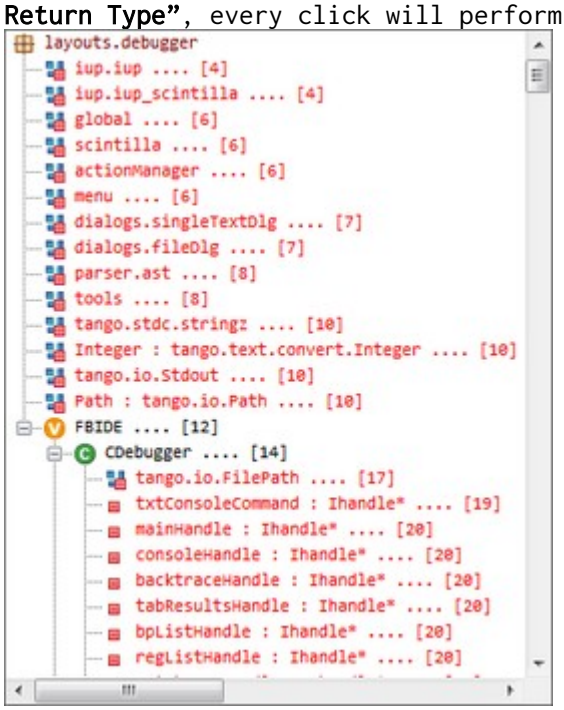


Form left to right with five buttons:
(1) Collapse.
(2) Show the Parameters / Return Type(Type).
(3) Show line Number
(4) Search word from head. (for search bar)
(5) Refresh the parser of active document. (parse text, no need save file)
(6) Hide Outline Panel.

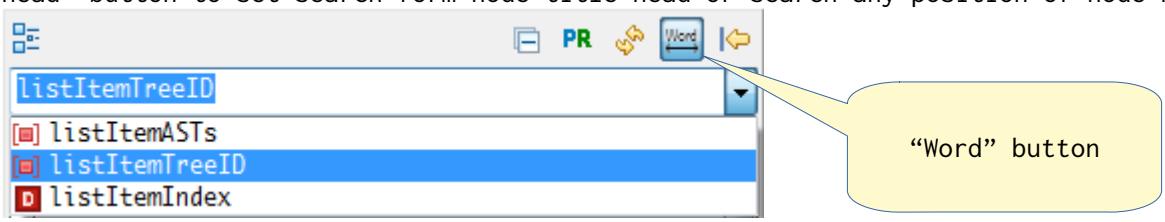
Click the "Collapse" will collapse the nodes at same layer.

"Show the Parameters / Return Type(Type)" had 4 type: All / Only Parameters / Only Type / None, every click will perform every existed outline of documents.

“Show line Number” will show line number of ever nodes, like “Show the Parameters / Return Type”, every click will perform every existed outline of documents.



If we want to search the name of node, type word in search bar then press “Alt + DOWN"(Linux also support “Enter” to search) or click the dropdown button(Linux unsupported), the search rule is always case insensitive, and we can press the “Search word from head” button to set search form node title head or search any position of node name.



If we use keyboard to trigger the matched list, we can move the light bar by press “UP” and “DOWN” then press “ENTER” to sure, the document also jump to selected node’s line number.
If using mouse, just scroll to the list item then click to select.

poseidon parser start working while loading file(*.d / *.di)(poseidonD) or (*.bas / *.bi)(poseidonFB), after produce a parsedTree, next poseidon will search the D_IMPORT nodes at second layer(first children layer of D_MODULE), then load the D_IMPORT files and parse again, but for efficive, just parse the D_IMPORT files of the this main D_MODULE, not others D_IMPORT of D_IMPORT, the source and others parsedTree will put into memory.

The D compiler import paths were defined in sc.ini / dmd.conf file and also in “Import Paths” of project, so poseidonD will search files by them.

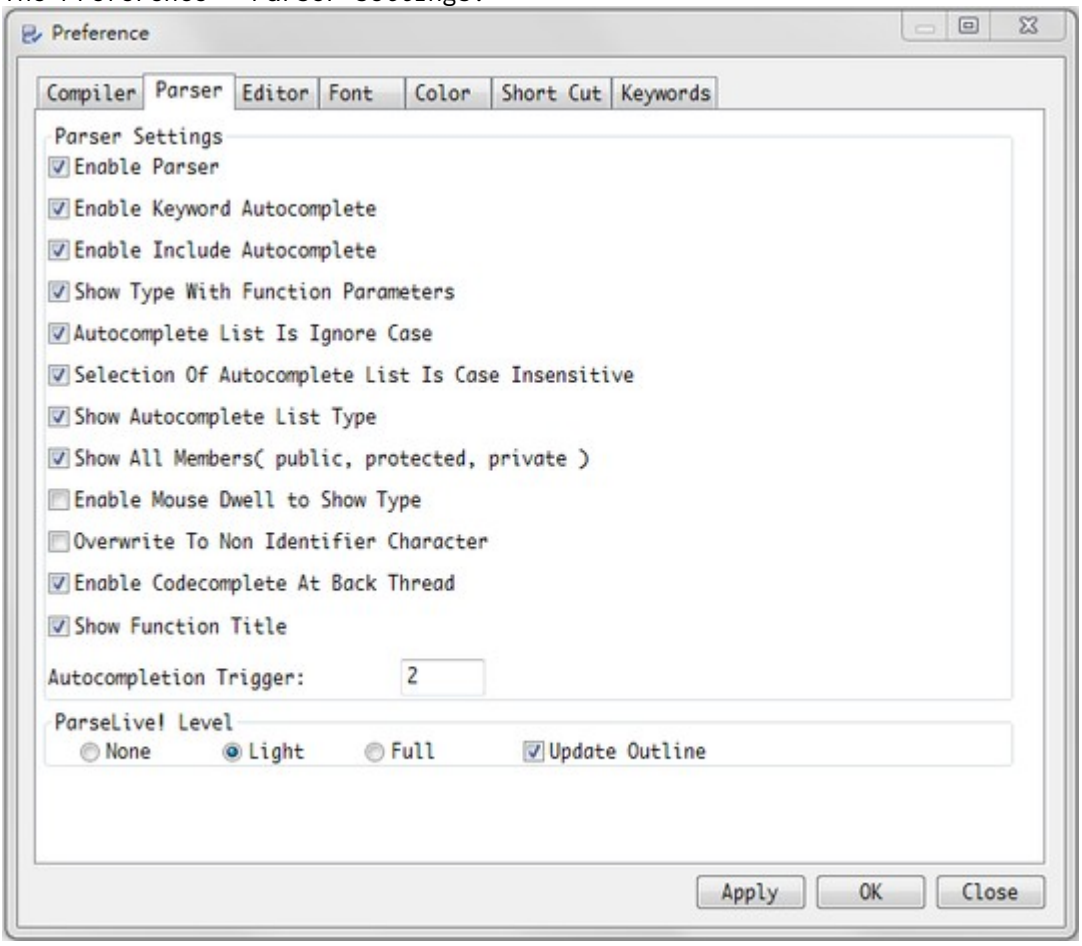
The freeBASIC has it’s own include rule--
(1) Relative from the directory of the source file
(2) Relative from the current working directory
(3) Relative from addition directories specified with the -i command line option
(4) The include folder of the FreeBASIC installation

In fact, D language will a object.d / di, poseidonD will try to load it and parse while launching, in poseidonFB, it will load and parse the FB_BuiltinFunctions.bi(in settings folder) to get keywords calltip.

Auto Complete

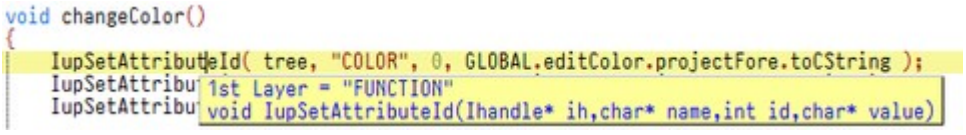
poseidon support 3 type auto complete:
(1) Keyword Complete
(2) Include Complete (Import Complete)
(3) Code Complete

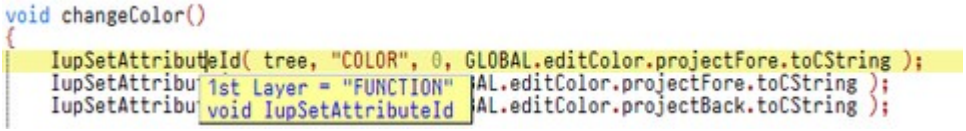
The Preference - Parser settings:



The **“Enable Keyword Autocomplete”** is work whatever the “Enable Parser” was On / Off, we can also goto the “Keywords” tab in “Preference” dialog to edit our keywords.
The (2)Include Complete and (3)Code Complete are work by the first option -- **“Endble Parser”**, ever **“Enable Include Autocomplete”** was ON but **“Enable Parser”** is OFF, the Include Complete isn’t work.

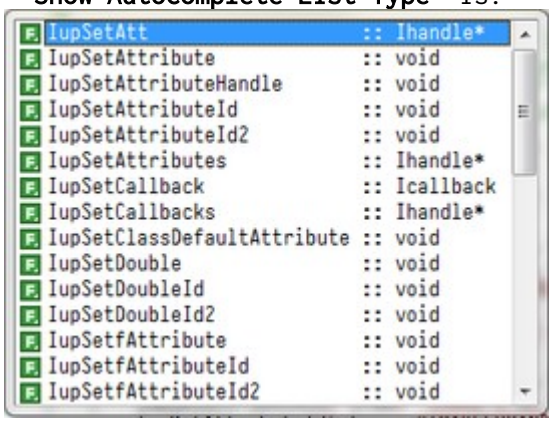
Diifference of **“Show Type With Function Parameters”** are:

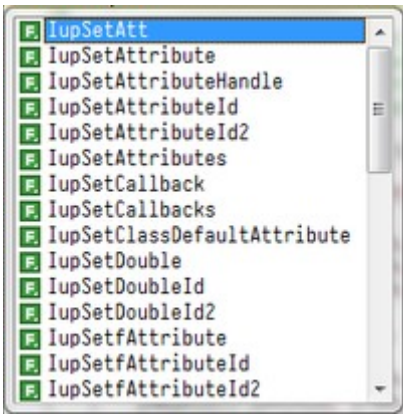




The **“Autocomplete List Is Igore Case”** and **“Selection of Autocomplete List Is Case Insensitive”** are all about case sensitive. The two options are for other language(poseidonFB) and personal habit.

"Show Autocomplete List Type" is:



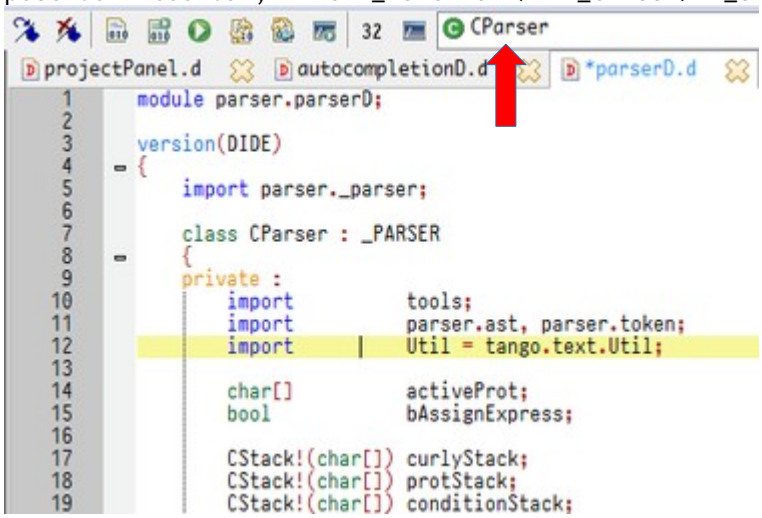


poseidon will sort lists and remove the duplicate ones, so with type or not sometimes will show different result. (In fact, different icons mean different ones)

The **"Show All Members"** will Show / Hide the private member, if "OFF", poseidon still check if the member is friend or the same parsedTree brance.

The **"Enabled Codecompletion At Back Thread"** is "ON", poseidon will create two threads to perform autocomolete and calltip at background, it prevents the IDE idle waiting for the the autocomolete / calltip results to speed up, but sometimes it is buggy, try to "OFF" if the result is strange or the system is unstable.

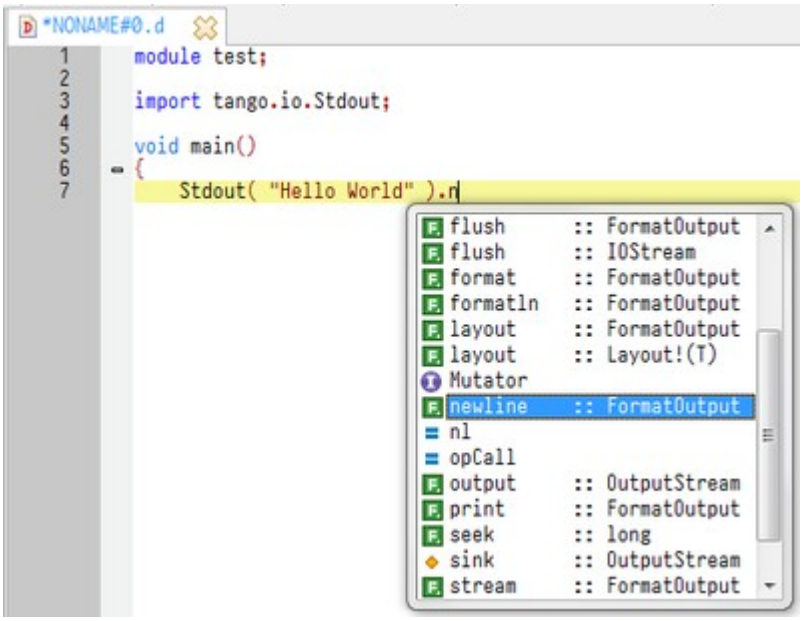
The **"Show Function Title"** will show the active block of document now at right side of poseidonD toolbar, like D_FUNCTION / D_CLASS / D_STRUCE etc.



The **"ParseLive!"** Will dynamic parse what we key-in and add to parsedTree:
(1) None = Turn off.
(2) Light = Parse one line.
(3) Full = Parse one block. (It's slow)
If "Update Outline" is "ON", the tree of outline will change by new parsed result.
The "ParseLive!" is not very accurate, use "ReParse" to parse the document again if necessary.

So we can luanch the poseidon and set the compiler's path, create a new file, type some code, the auto complete will trigger without other settings.

The parser of poseidon is for produce the parsedTree as the first goal, the parsedTree is for autocomplete / calltip use and the parser isn't very accurate, it pass the unittest, without if / while / for / foreach / foreach / switch blocks, we can watch the outline, like below image, we get `result : char[]` x2, but it will not effect the auto complete working.



Calltip

If “Enabled Parser” was “ON”, it works.

Calltip have three hidden settings of change color, edit the editorSettings.ini, goto [COLOR] block and check three items:
calltipFore
calltipBack
calltipHLT

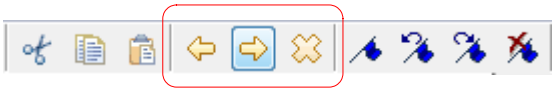
assign them with = R G B.

Manual AutoComplete:

If we press ESC key or move the cursor to cancel the autocomplete list, no need add / remove any word, press the default short-cut: Ctrl+Q to show the list or calltip again.

Goto Definition:
Goto Member Procedure(poseidonFB):
Goto Top Definition(poseidonD):

When **“Enable Parser”** is “ON”, we can jump to the declare of variables / functions, because the parsedTree had created, it easy to implement.
We can Jump to the locate of the declares, but how to back to origin? Use “Navigation Cache” can backward / forward look.



Except for click the icons, we can also use the short-cuts or ALT + mouse “LEFT” / “RIGHT” click to perform.

Include Levels:(poseidonFB)
This options only existed in poseidonFB:

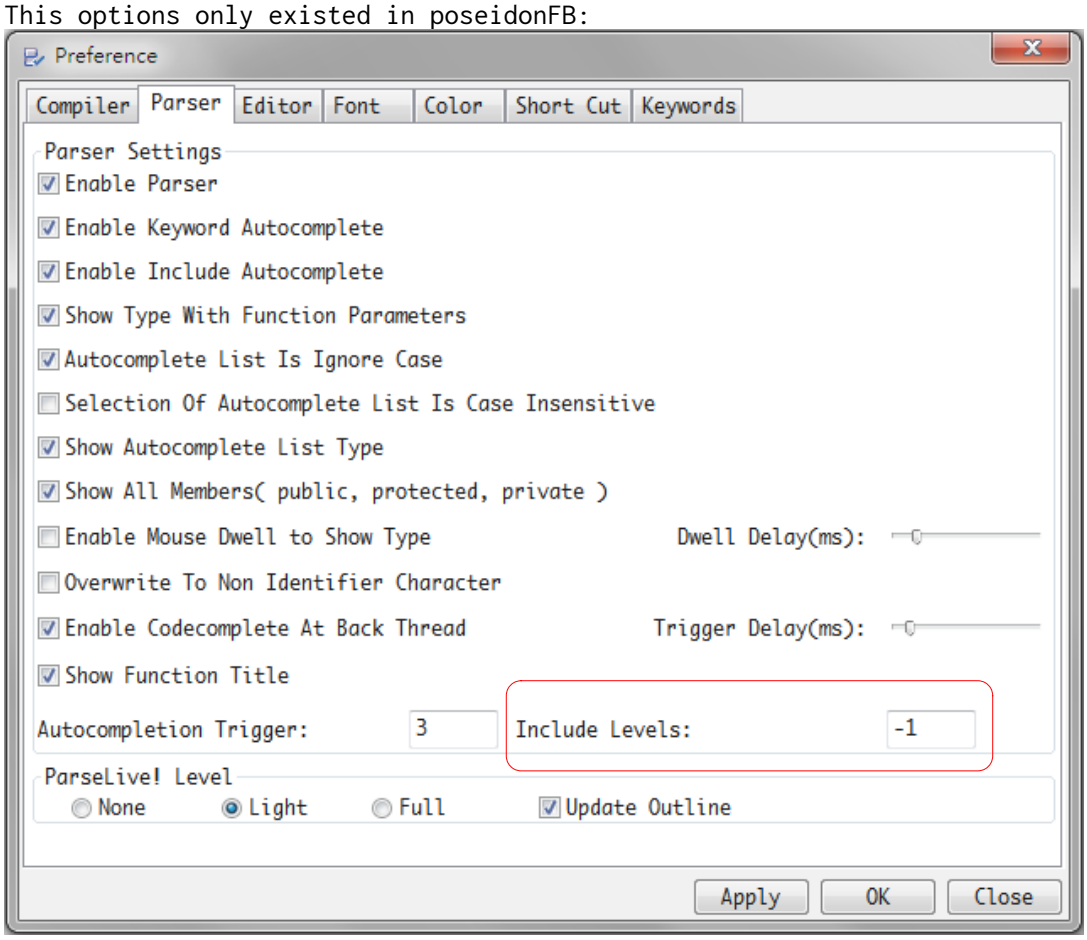
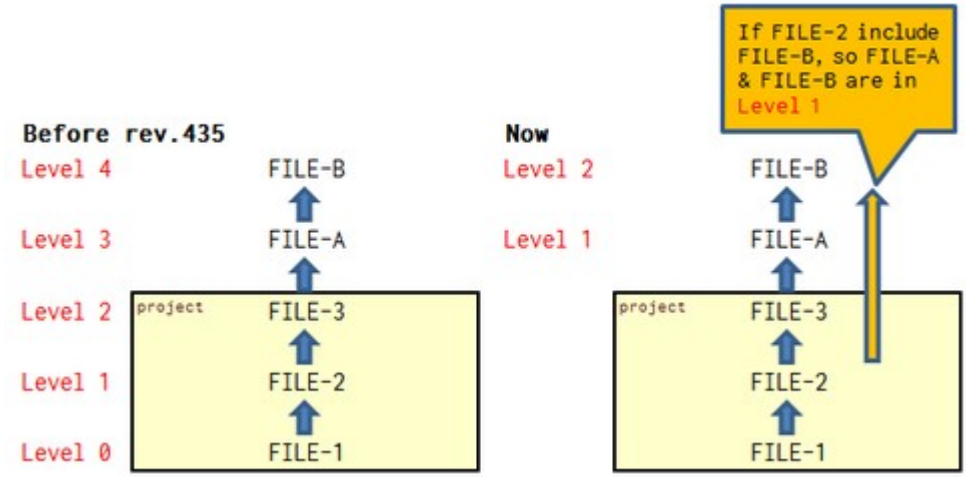


Image when our code include "window9.bi"(Gui Library win-dow9,https://sourceforge.net/projects/guiwindow9/), window9.bi also include many *.bi, the different *.bi also include more *.bi, the parsedTrees of poseidonFB will grow to very huge and we need search and compare every nodes to perform codecomplete, the speed will get slower, so I design an option -- "Include Levels" to avoid parsing over and over *.bi

There is some change about poseidonFB, >=reversion 0.435, all files in project are level 0 to avoid when level = 1, FILE-1 can't use FILE-3 even at same project; the in-clude rule as bottom pic, when "Include levels" = 1, FILE-2 and FILE-1 can show the parseTree of FILE-B, but FILE-3 only show FILE-A.

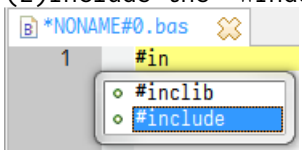


Set the levels = -1, there is no limit that poseidonFB will include all files.

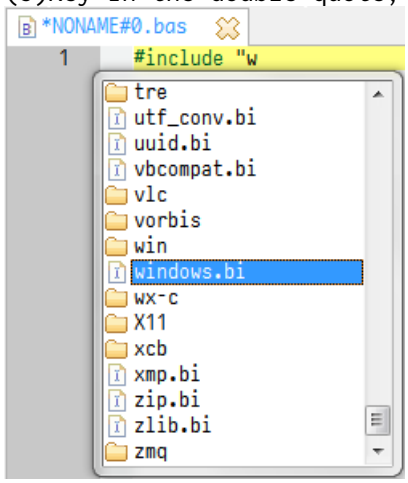
Autocomplete Example: Show a MessageBox

(1)Launch poseidonFB, create a new file.

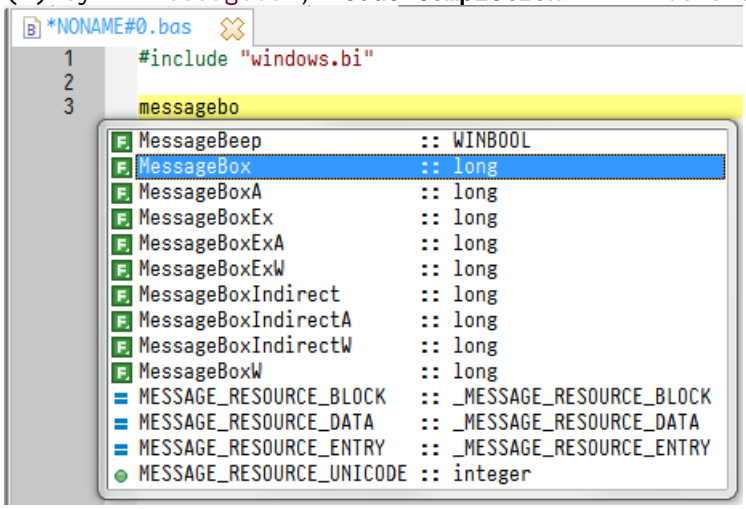
(2)Include the “Windows.bi”, key in #in, the **“Keyword Autocomplete”** will be shown:



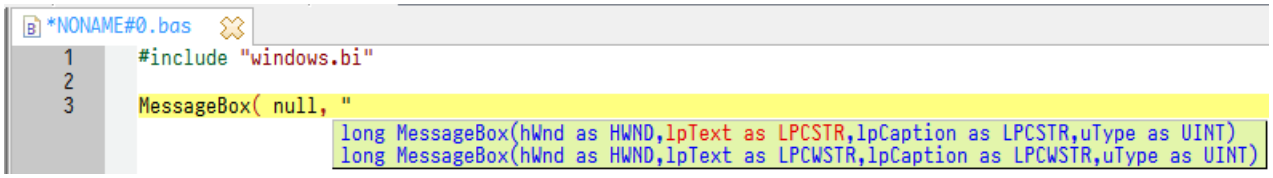
(3)Key in the double quote, the **“Include Autocomplete”** will be shown:



(4)key in messagebox, **“Code completion”** will be shown:



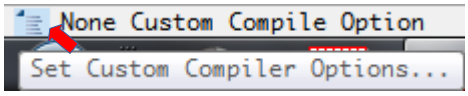
(5)After key in the left parenthesis, the **“Calltip”** will be shown:



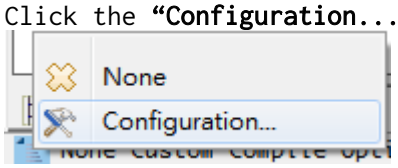
(6)Quick Run

Custom Compiler Options

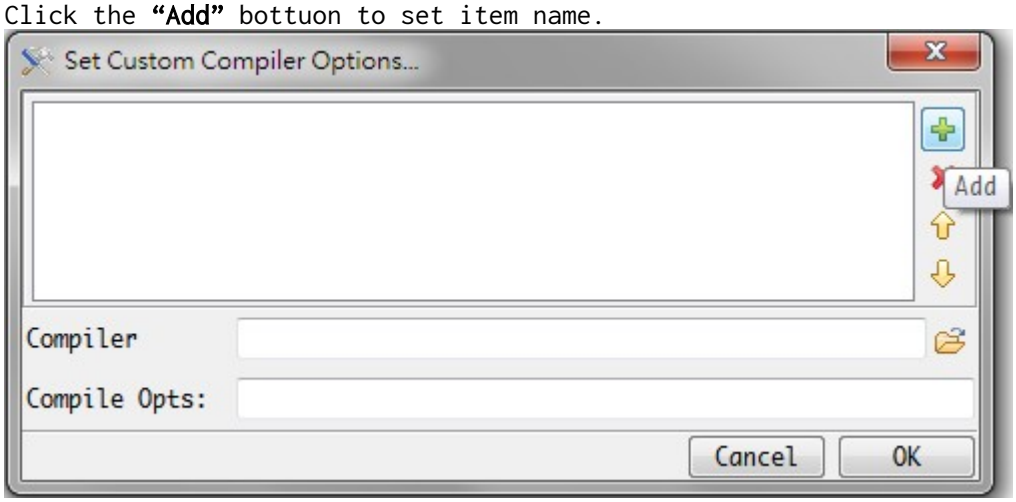
This can use pre-setting Compiler and it's options quickly, at first, left/right-click on left-bottom button:

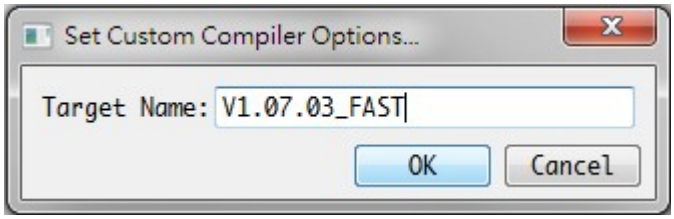


Click the **"Configuration..."**:

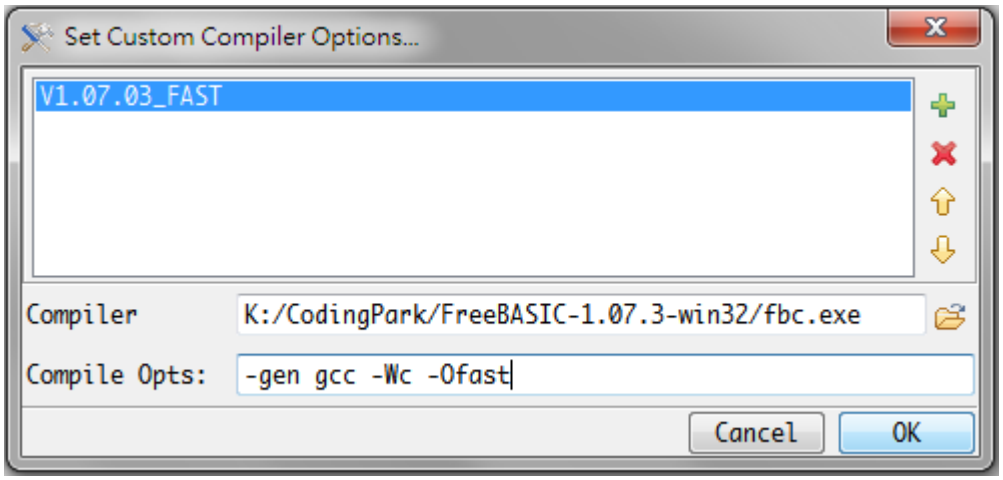


Click the **"Add"** bottuon to set item name.

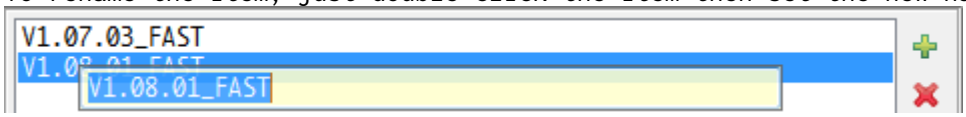


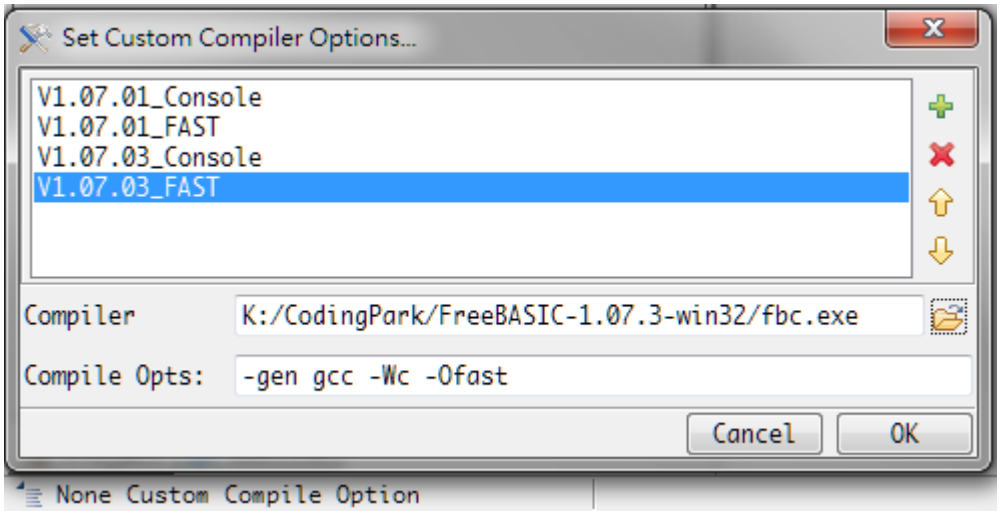


Set the Compiler path and it's options, click **"OK"** or click the **"Add"** to continue creating a new item.

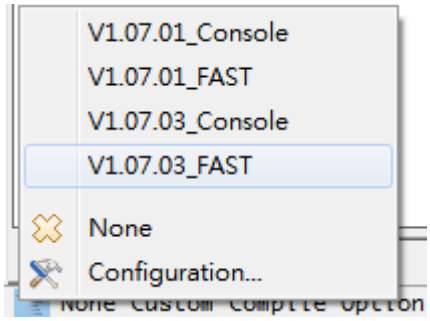


To rename the item, just double-click the item then set the new name.

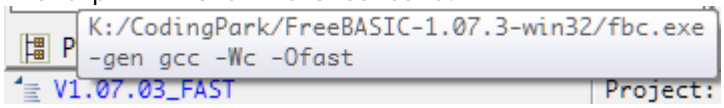




Click the left-bottom button again, it will show the items list and select one of them.



The tip will show it's content:

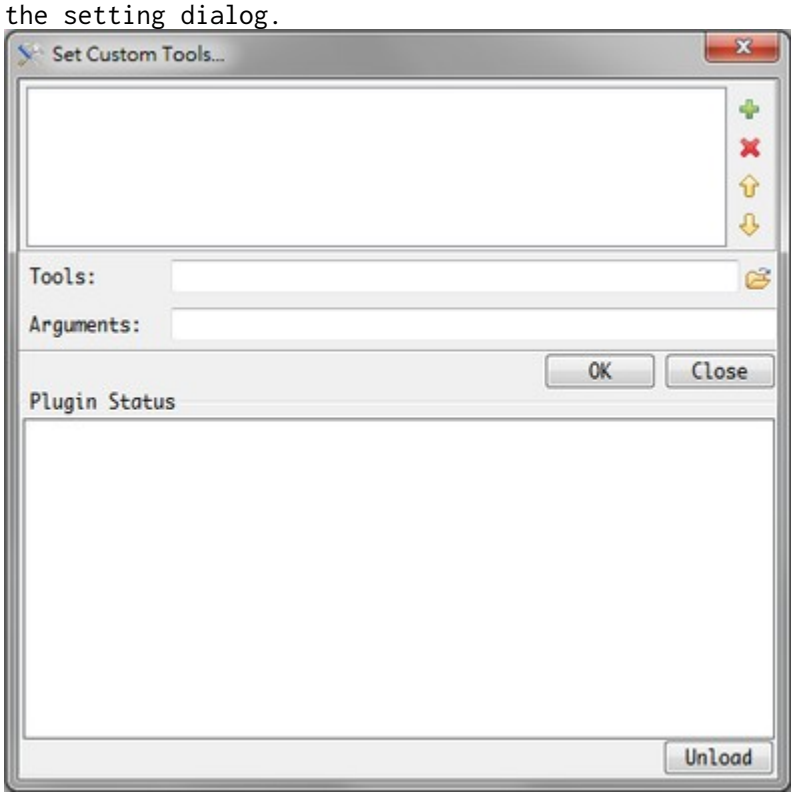


Note:
The sequence of selected compiler path is (1)Right-click on icon (2)Custom Compiler Option... (3)Compiler path in 'Project Properties'(if project used) (4)Compiler path in 'Preference'

Custom Tools for poseidon

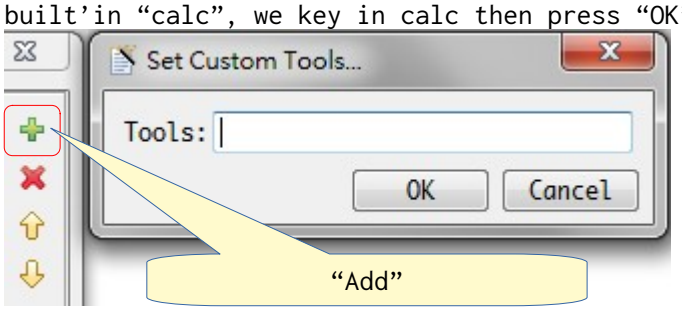
poseidon supports custom tools both executable programs(.exe files) or Plugins(*.dll / *.so), the plugins can access poseidon's IUP framework data (including scintilla) and also expands the framework's capabilities.

To setup custom tools, click 'Menu' > 'Options' > 'Tools' > 'Set Custom Tools…' to open the setting dialog.

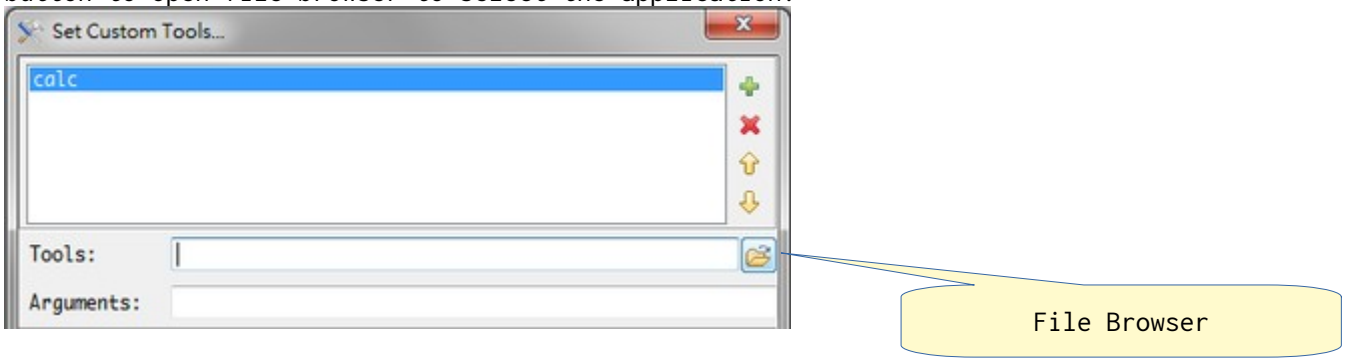


Set External Application:

At first, click the **"Add"** button, set the tool's name, for example: set the windows and built'in "calc", we key in calc then press "OK".

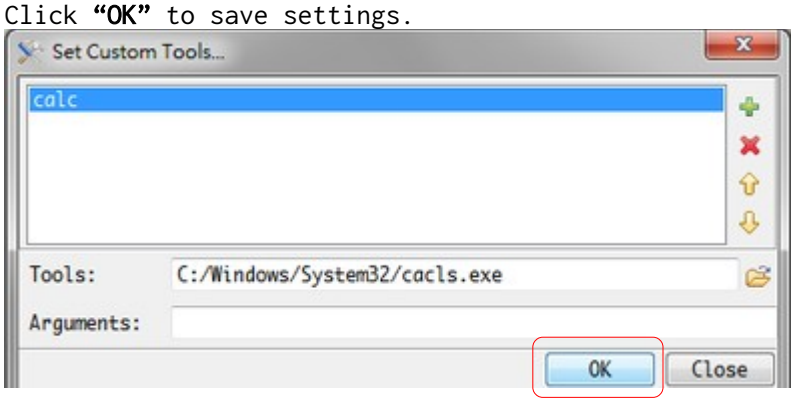


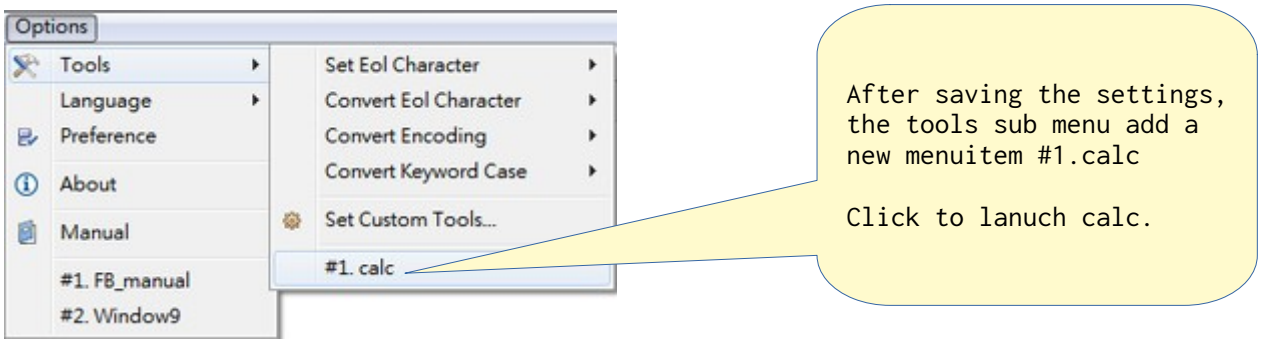
In the "Tools:", we need set the application's fullpath, also we can click the **"Open"** button to open file browser to select the application.



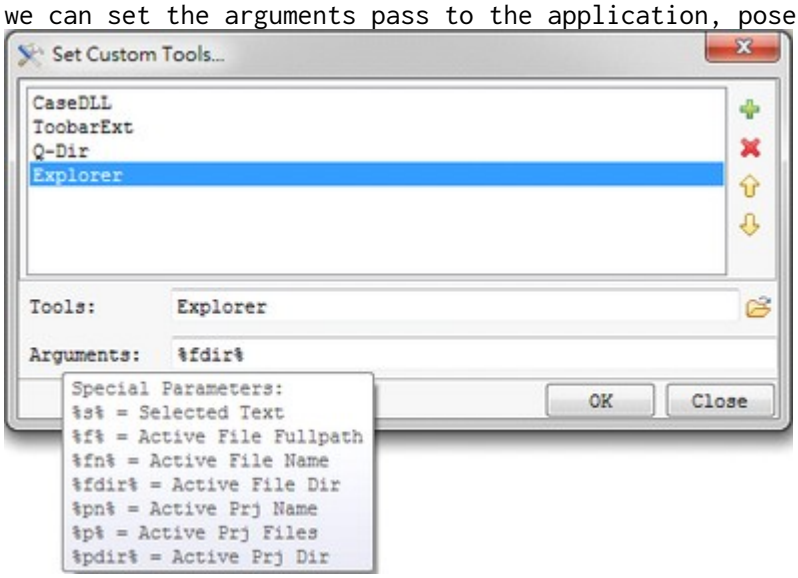
Of course, we can key in calc directlly.

Click **“OK”** to save settings.





Poseidon support 12 custom tools and the short-cut default is Shift + ctrl + F1 ~ F12, we can set the arguments pass to the application, poseidon has some special parameters:



Special parameters list:

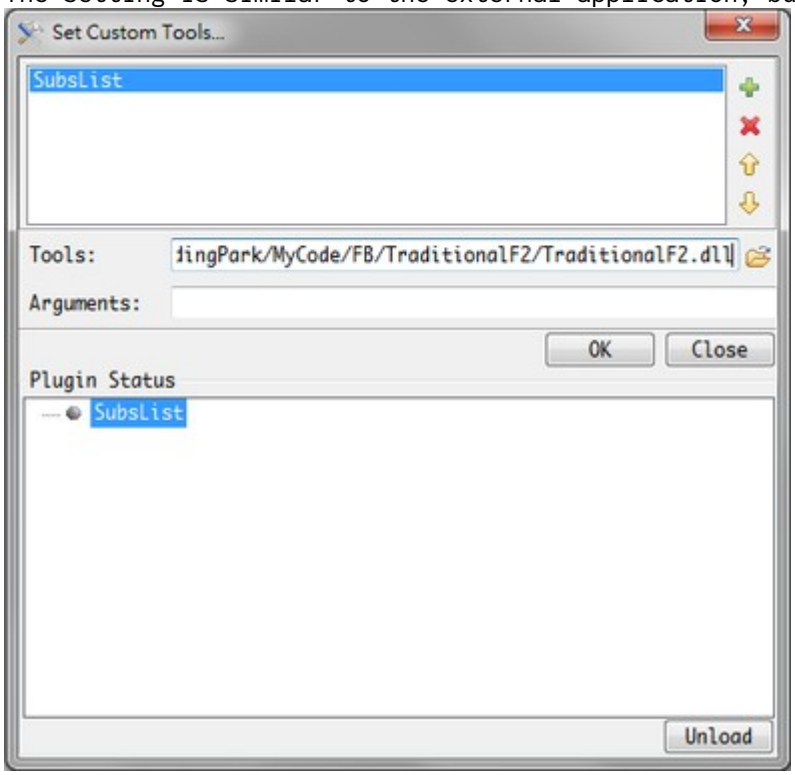
%s%	Selected Text
%f%	Active File Fullpath
%fn%	Active File Name
%pn%	Active Prj Name
%fdir%	Active File Dir
%p%	Active Prj File
%pdir%	Active Prj Dir

For above picture example, launch the external application will run windows explorer to file dir.

Plugin:

poseidon uses IUP framework, the biggest highight is built'in scintilla supprot - it's iup_scintilla, we can make IDE more easy by using the iup_scintilla.

The setting is similar to the external application, but without Arguments:



The **"Plugin Status"** will show the loaded plugins, we can release the plugin from memory by **"Unload"** button.

The design of poseidon plugin is also using the iup_scintilla(IUP) functions to access with the poseidon IUP objects, eq: **IupGetHandle** or **IupGetDialogChild**, to get the poseidon main dialog, we use:

```
dim as Ihandle ptr POSEIDON_HANDLE = IupGetHandle( "POSEIDON_MAIN_DIALOG" )
```

if we want get active document handle, we use:

```
dim as Ihandle ptr _tabsHandle = IupGetDialogChild( POSEIDON_HANDLE, "POSEIDON_MAIN_TABS" )
dim as Ihandle ptr _sci = cast( Ihandle ptr, IupGetAttribute( _tabsHandle, "VALUE_HANDLE" ) )
```

or

```
dim as Ihandle ptr _sci = IupGetFocus() ‘ Need before main dialog show
```

When we get the handle, we can do many works with the document.

The plugin program need a entry function for poseidon call at least, it is:

```
extern "C"
    declare sub poseidon_Dll_Go alias "poseidon_Dll_Go" ( _dllFullPath as zstring ptr )
    declare sub poseidon_Dll_Release alias "poseidon_Dll_Release" ()
end extern
```

The **poseidon_Dll_Go** function with a parameter, we can save the settings of plugin by it, the **poseidon_Dll_Release** will call when plugin unload, we can use it for release somethings, save somethings…

I’ve commit a project at bitbucket, there are some examples of plugins, please refer: https://bitbucket.org/KuanHsu/poseidon-plugins/src/master/

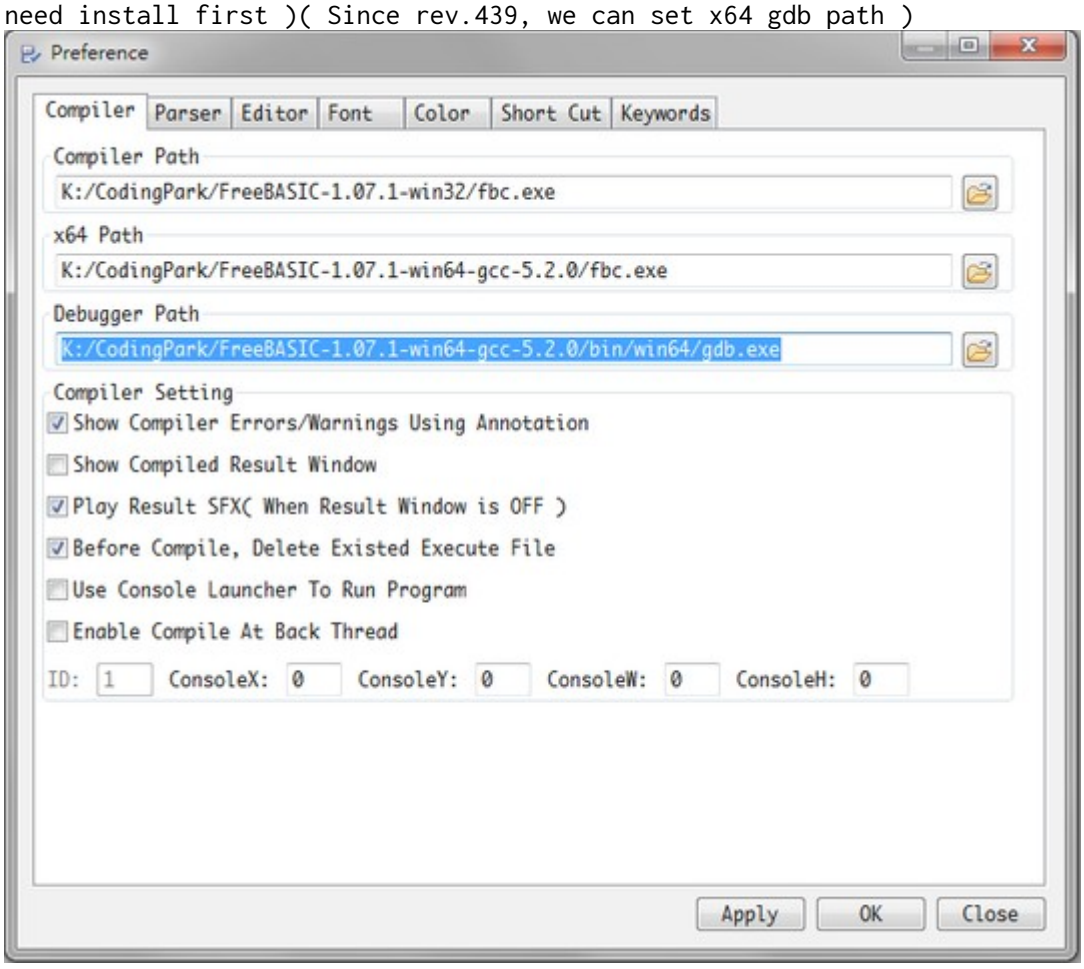
Debug: (poseidonFB only)

I'm sorry that I don't have much experience about debug, I'll do my best to describe how's work in pictures.

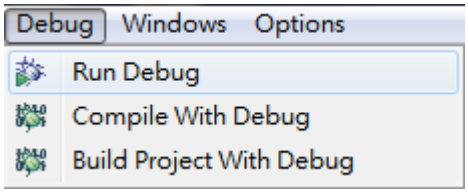
poseidonFB use gdb to debug, offical freeBASIC's pack include 32bits gdb nor 64bits(to debug 64 bits program need 64 bit debugger), we can download from https://www.freebasic.net/forum/viewtopic.php?f=6&t=27859 and install the 64bits gdb.

In my test, it works on:
(1) freeBASIC 1.07.1, Win32 with GNU gdb (GDB) 7.6.1
(2) freeBASIC 1.07.1, Win64 with GNU gdb (GDB) 9.1 (srvaldez built)
(3) freeBASIC 1.07.1, linux mint 18 64bit with GNU gdb (Ubuntu 7.11-0ubuntu1) 7.11

Go to "Preference" -> "Compiler" to set the debugger path, in linux, just key in "**gdb**"(need install first)(Since rev.439, we can set x64 gdb path)

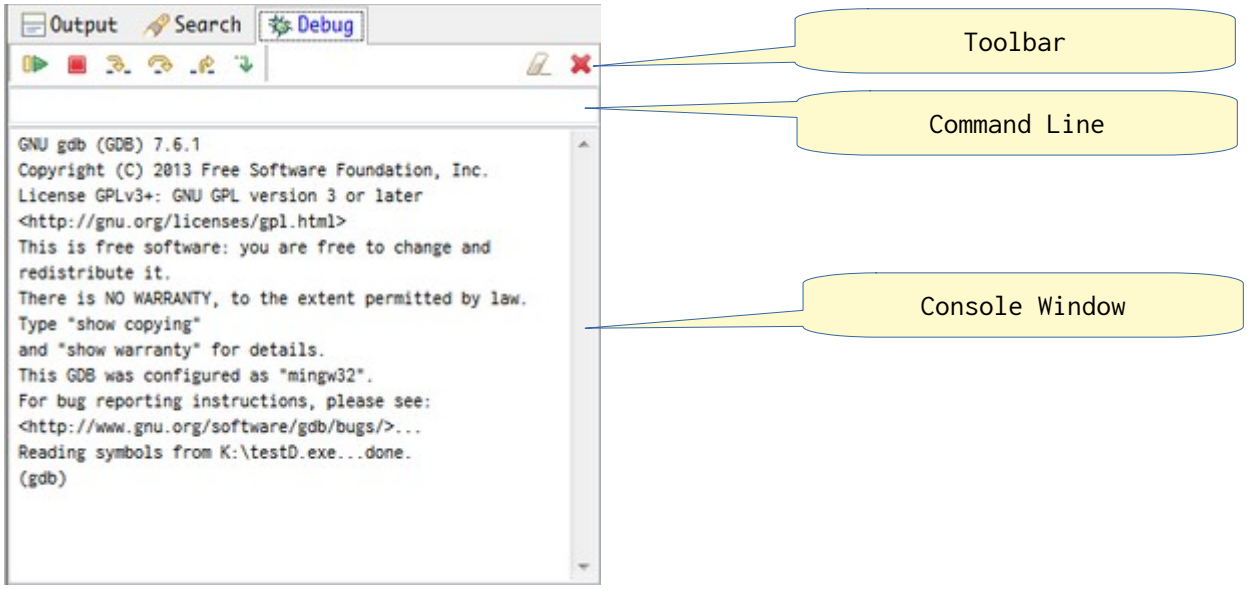


We need compile our sources with -g option to add debug information in our application, if we already cretate(used) a project, go to "Menu" -> "Debug" -> "Build Project With Debug", poseidon will apply all settings of project and add a -g option.

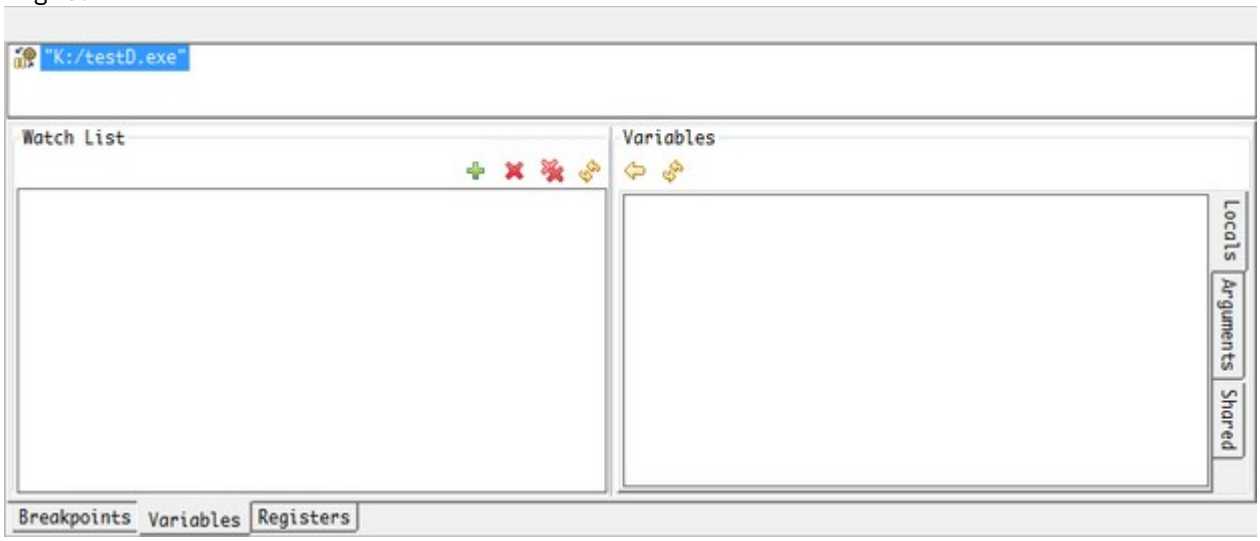


After compile the sources with -g option, click "Run Debug" to start debug, poseidon will execute the gdb in another thread, so we can operate the poseidon to set the break points / set args……

Left:



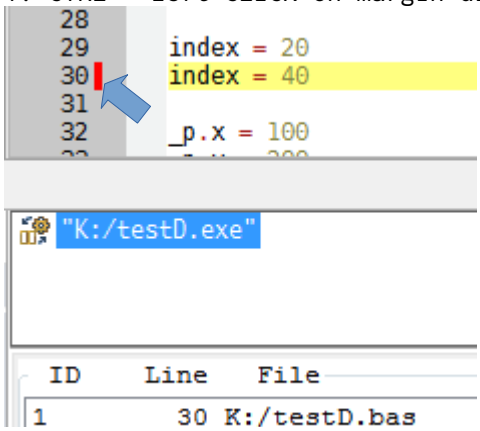
Right:



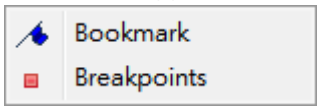
Breakpoint:

We have 2 methods to set bps:

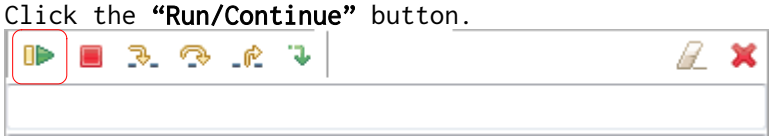
1. CTRL + left-CLICK on margin directly.



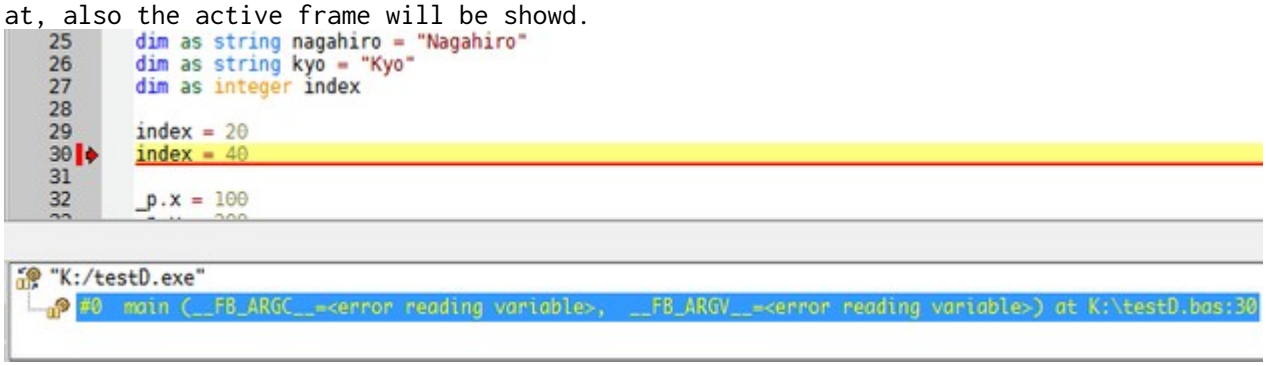
2. Right-CLICK on margin, select “Breakpoints” on popup menu or Right-Click on document while debugger had been run.



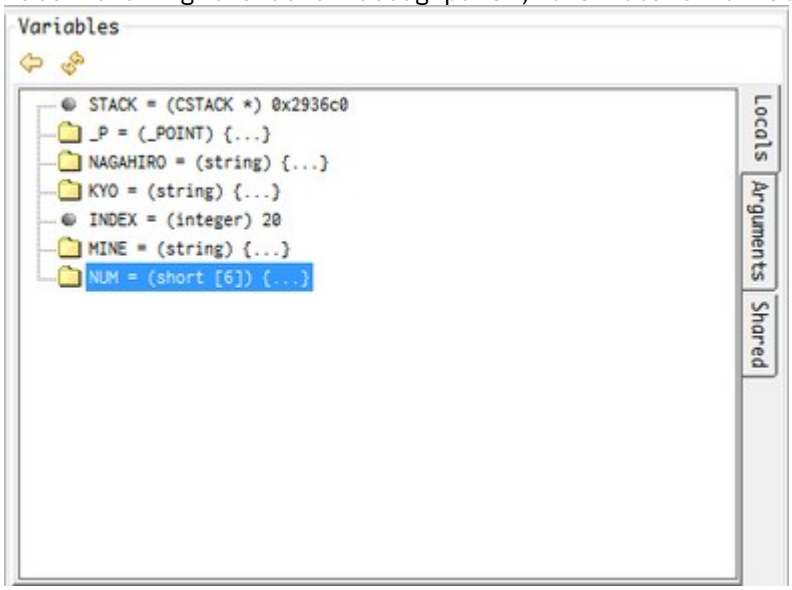
Click **“Run/Continue”** to run to breakpoint, if we need send args to application, Right-Click the **“Run/Continue”** button.



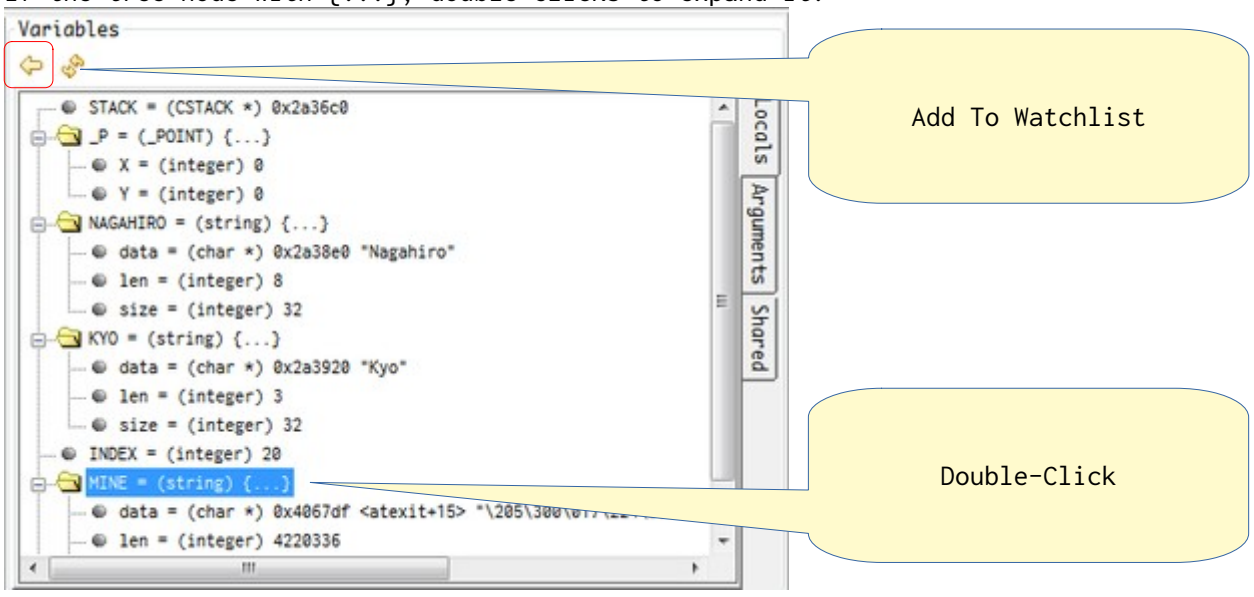
The margin will show the actual line with **red** arrow / **red** underline which debugger run at, also the active frame will be showd.



Watch the right side of debug panel, the locals variables list will be auto-updated.



If the tree-node with {...}, double-Clicks to expand it.

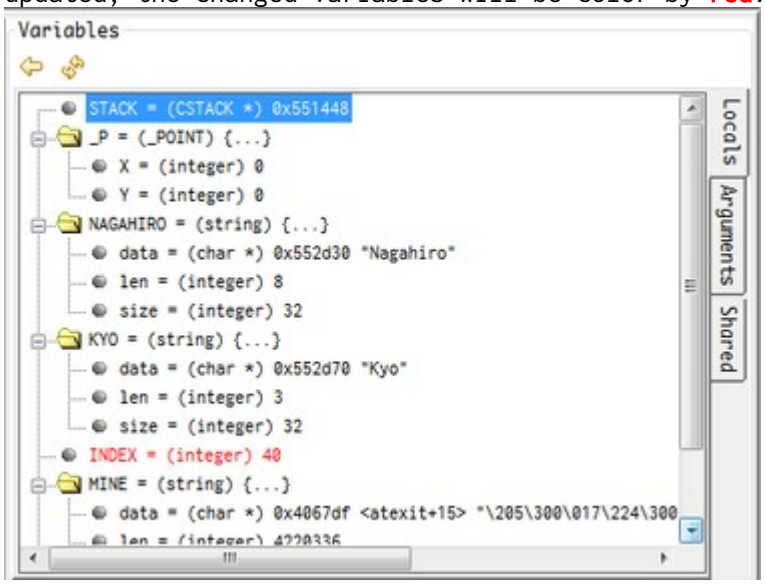


Toolbar:

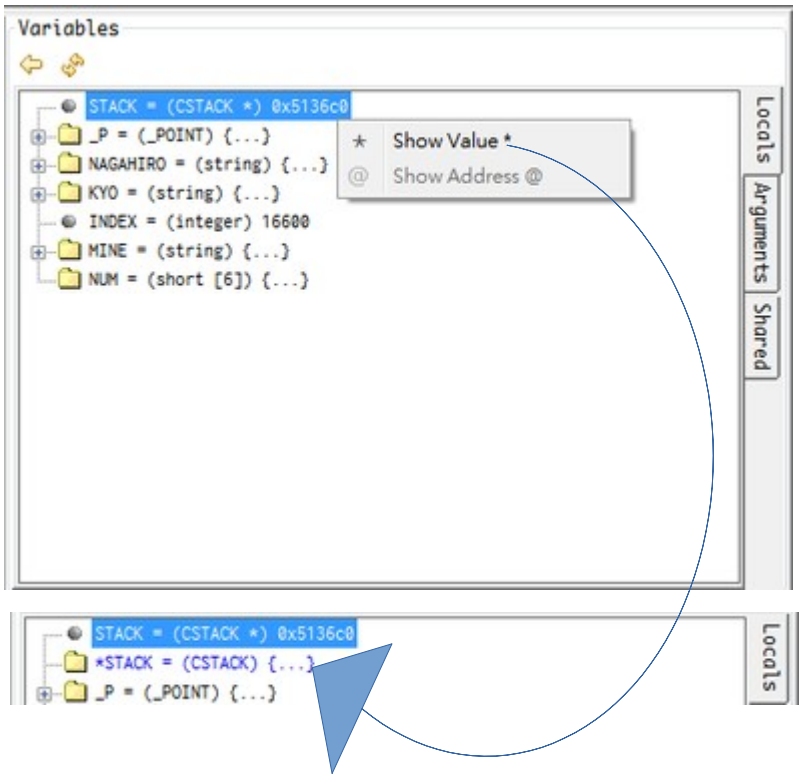
from left to right:
(1) Run/Continue
(2) Stop (kill)
(3) Step
(4) Next
(5) Return
(6) Until
(7) Clear the console window
(8) Terminate the debug thread

Command Line: Send the command to gdb directly.

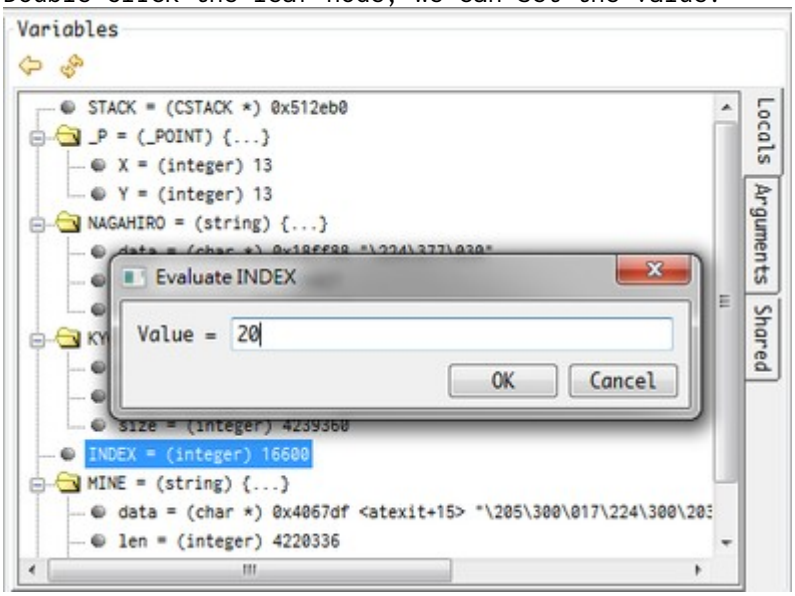
We press the **"step"** button to step by step debug, the value of locals variables will updated, the changed variables will be color by red.



We can also Right-Click on the node to show the **@address / *Value**:

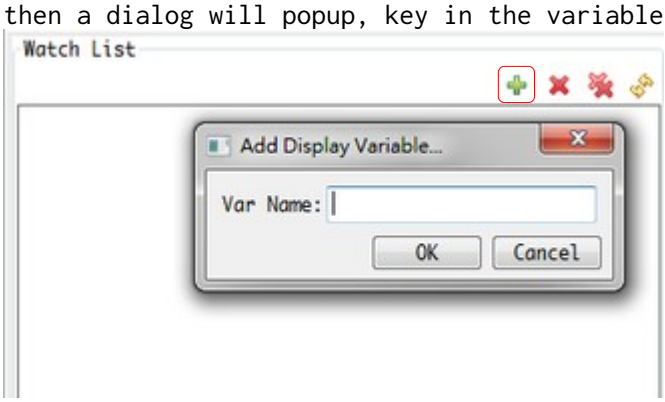


Double-Click the leaf node, we can set the value.

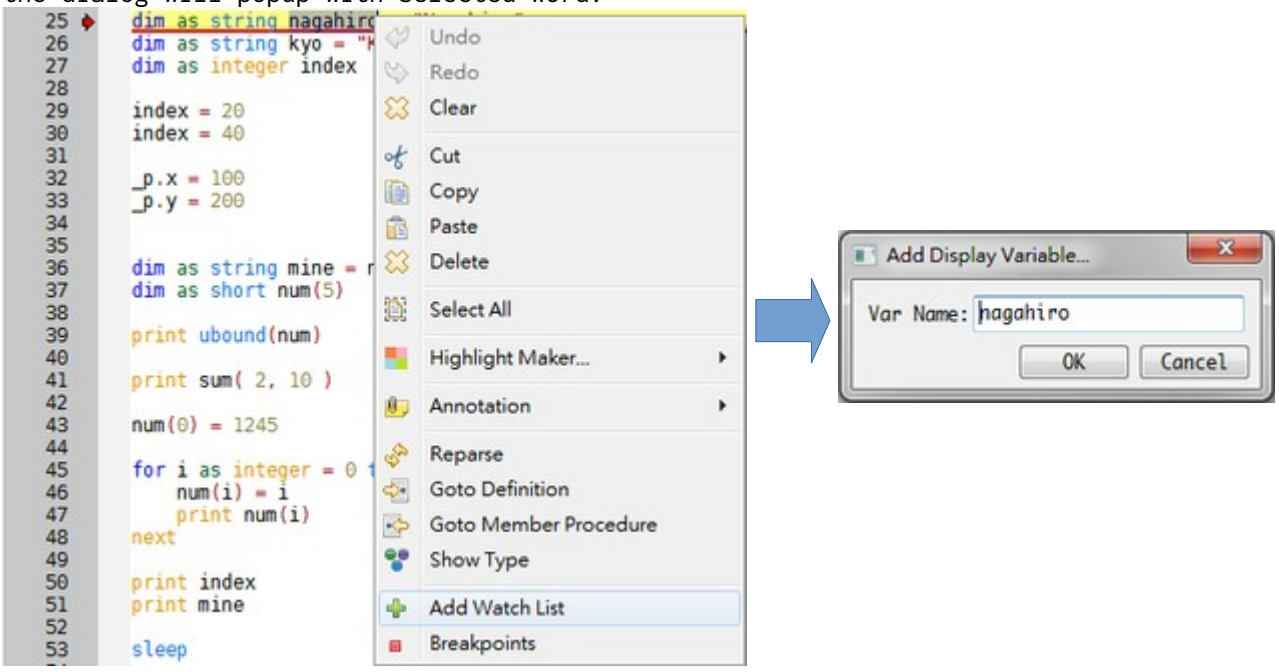


Watchlist:

There are several ways to add variable to watch list, the first is press Add button, then a dialog will popup, key in the variable name.



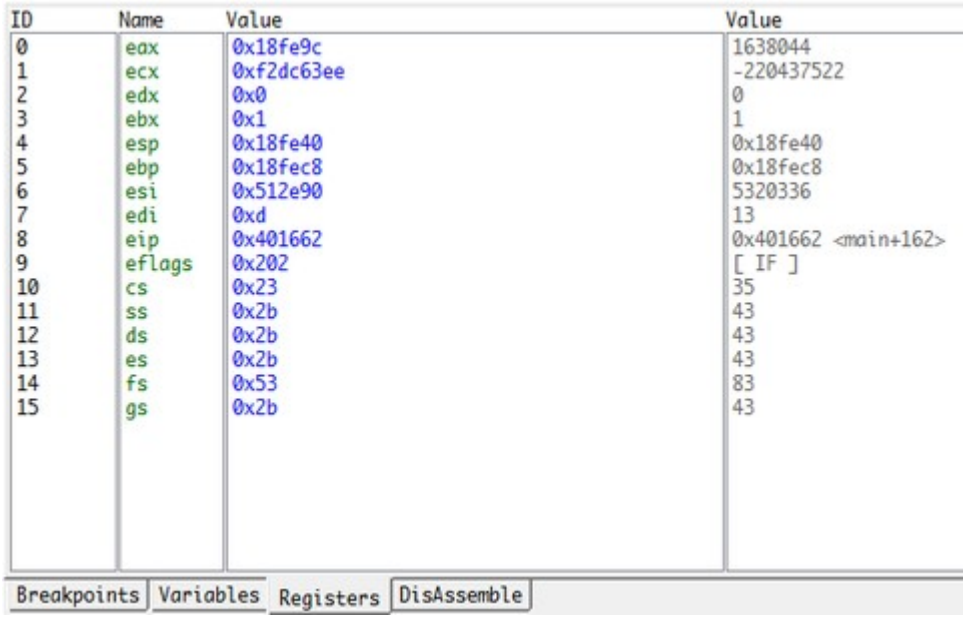
The second way is select a word of document, Right-Click and click **“Add Watch List”**, the dialog will popup with selected word.

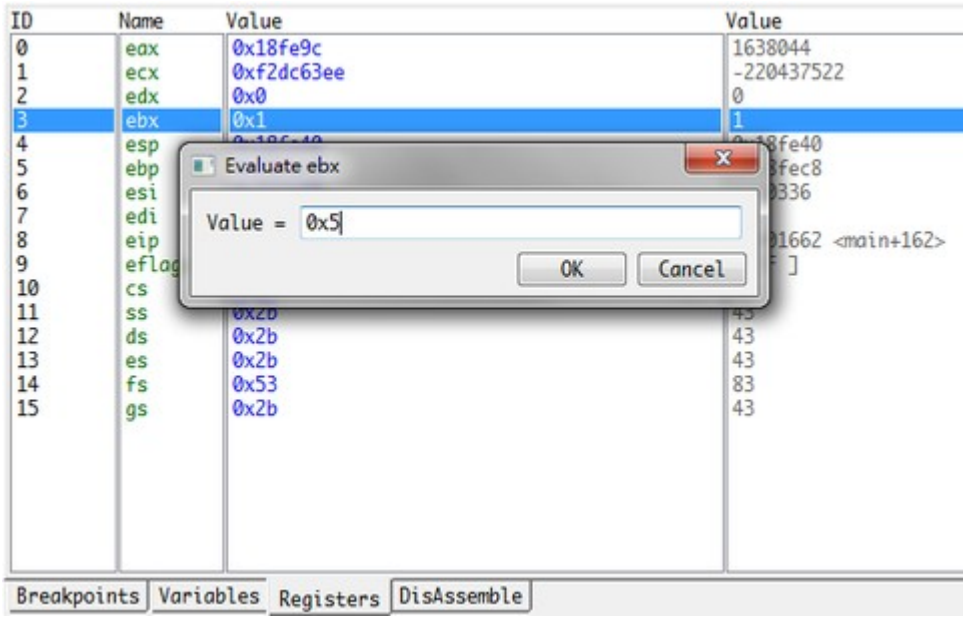


The third way is from Variable panel(use **“Right Arrow”** button).

Watchlist also support to show **@address / *Value** of variable, and poseidon will ask if add the new value to watchlist.

Registers:
The **“Registers”** panel will show the registers, also support evaluating the value, just Double-Click the item to do.

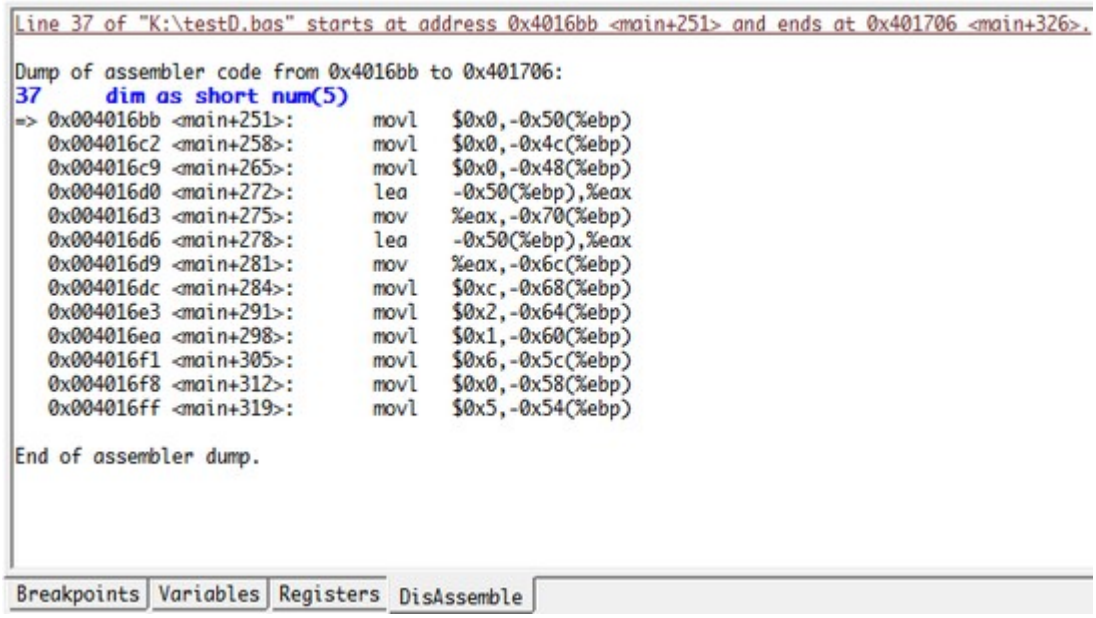




The last change isn’t color by red, because of there are some bugs about IupFlatList on IUP 3.27, the IUP 3.29 had been fixed but also got others bugs, so I stay at IUP 3.27 and waiting for **“Big”** version modified.

Although many bugs had been fixed at SVN, but we need build by ourselve…...

Disassemble:
poseidon will send **“disassemble /m startAddress,endAddress”** command to gdb and display asm code of current line.



Frame panel:
This panel will show which frame is working, the blue color word(not selection) is active frame, we can Double-Click to change frame, of course the watchlist and variables panel will also be update.