Magnet-DAQ Help

USER MANUAL

American Magnetics, Inc.

Quick Start

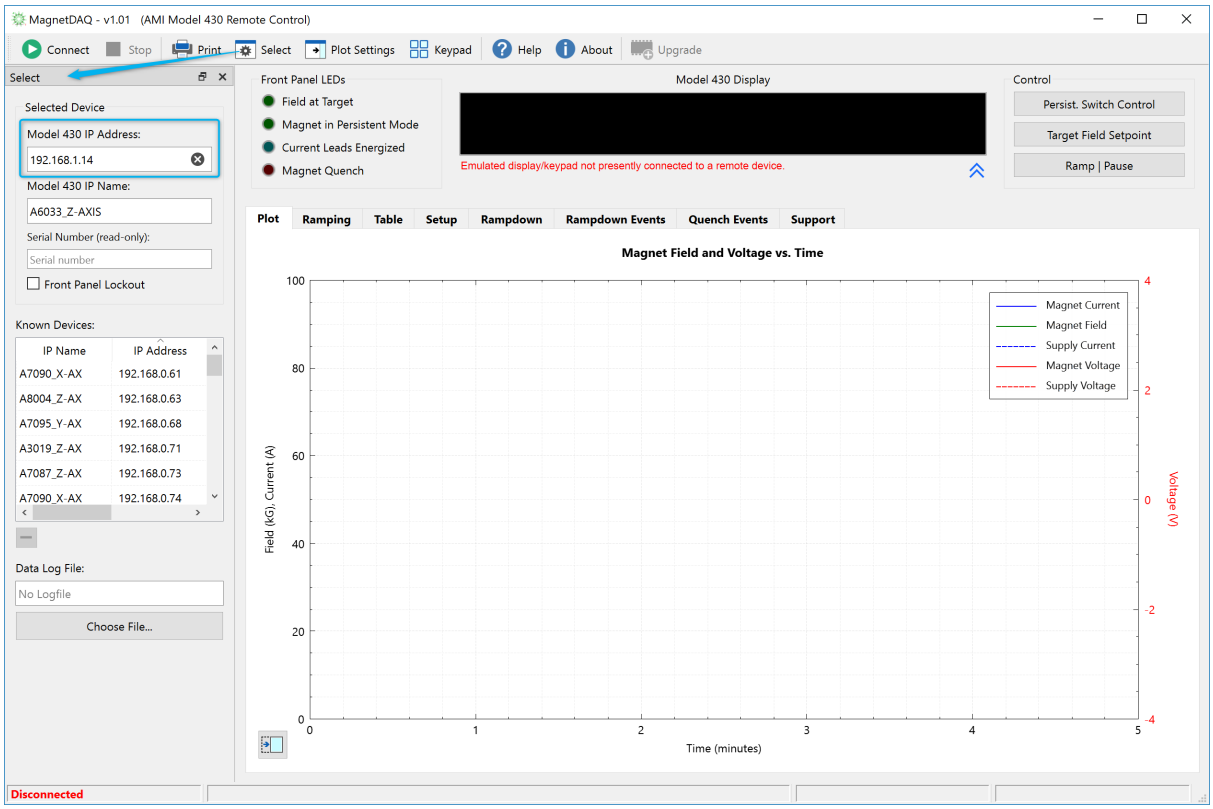
1 Quick Start

The Magnet-DAQ application is designed to be as simple to operate as possible but still provide a comprehensive interface for controlling the AMI Model 430 Programmer. The basic flow is as follows:

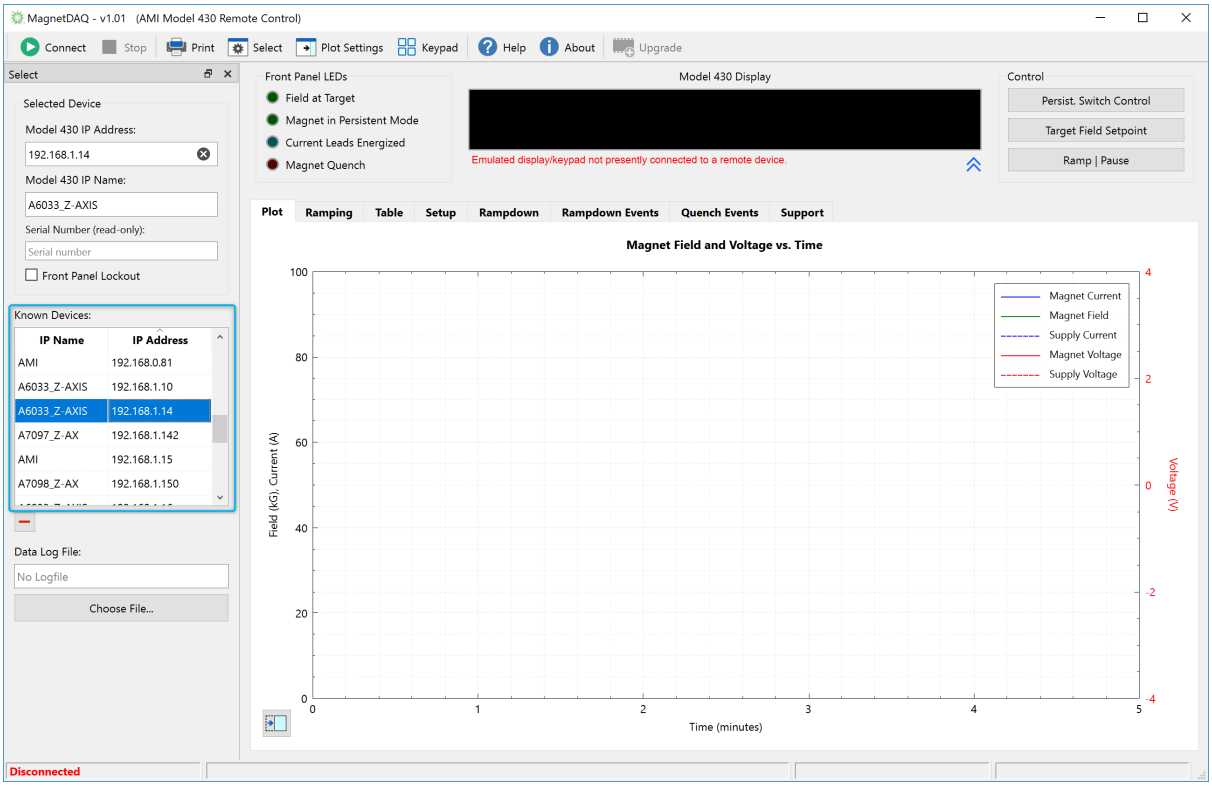
1) **Enter the IP address of the target Model 430.**
2) **Use the "Connect" button to initiate the connection and begin plotting data.**
3) **Control what data you wish to plot.**
4) **Use the remote Keypad and/or Ramping view to control the Model 430.**

Step 1: Enter the IP address of the target Model 430

Use the Select panel to enter the IP address for the target Model 430.

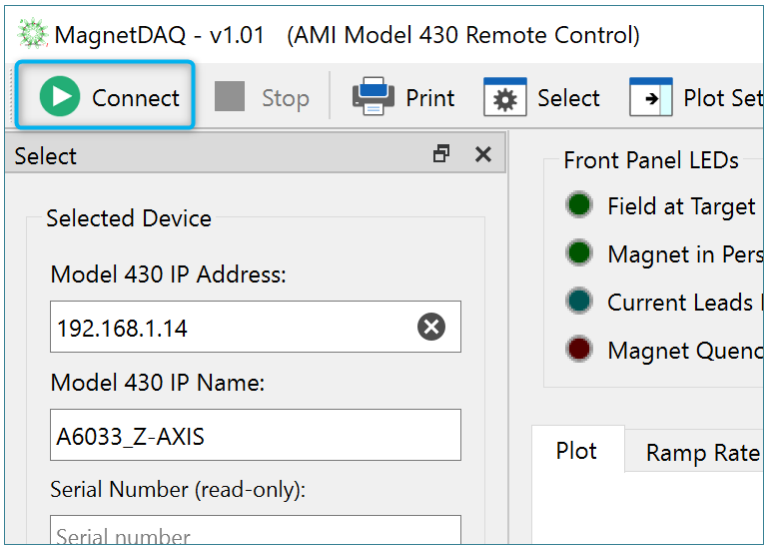


Alternately, if you have connected to the target Model 430 in a previous session, choose a known IP address from the "Known Devices" list.



Step 2: Use the "Connect" button to initiate the connection and begin plotting data

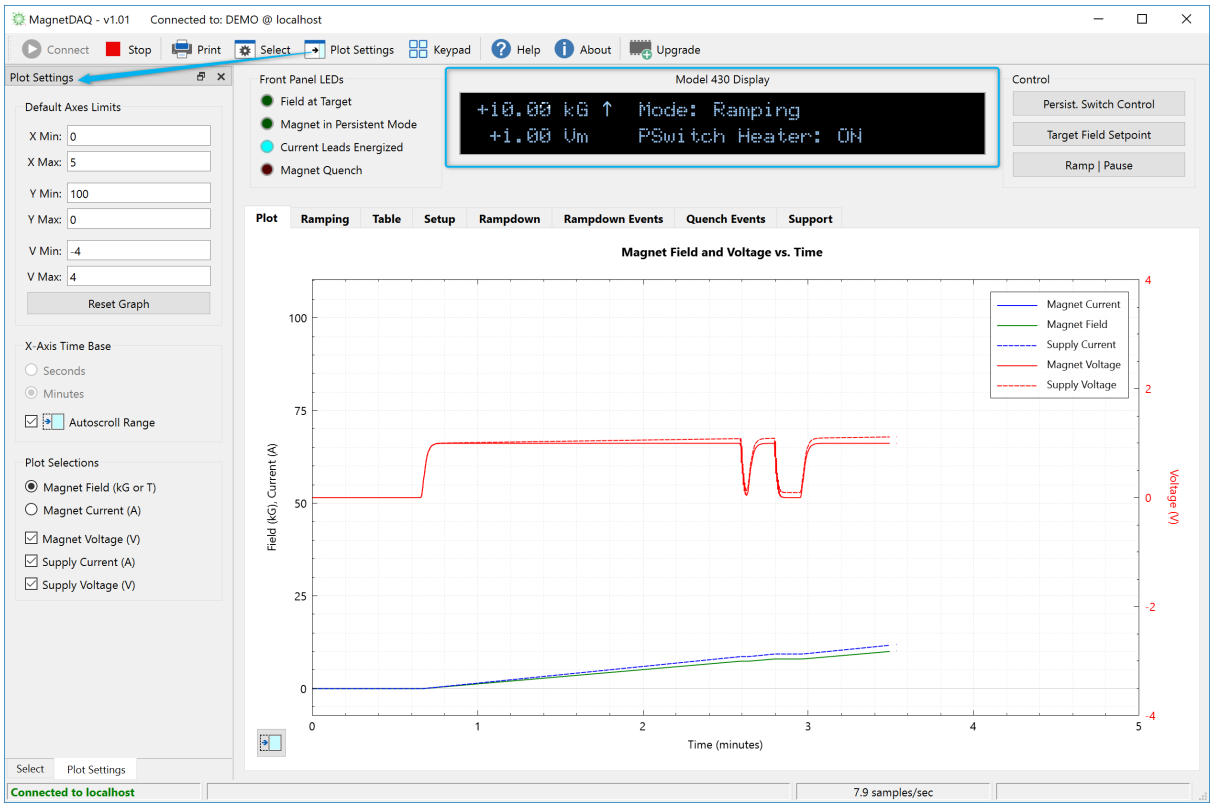
Use the **Connect** button in the toolbar to initiate the connection to the target Model 430. A progress panel will provide feedback during the connection and initial fetch of all the Model 430 settings. If the connection is successful, the dialog status area will display a "Connected to..." message and data acquisition and plotting of the data will start.



NOTE: Do you see a firmware upgrade notice? See this page.

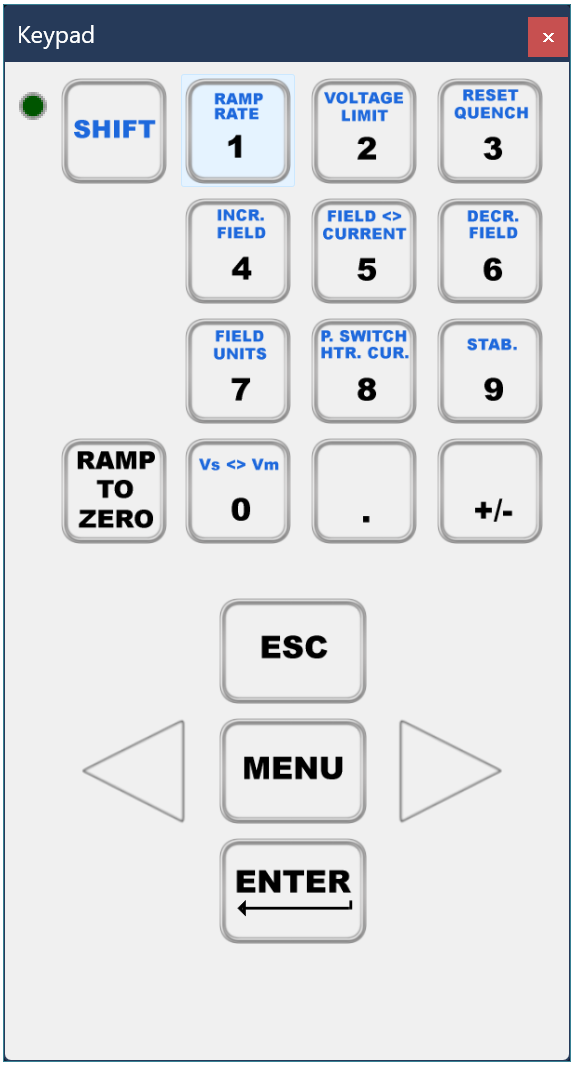
Step 3: Control what you wish to plot

Once an active connection is made, the Model 430 Display in the application will mirror the actual instrument display. Use the Plot Settings configuration panel to choose what data you wish to plot.

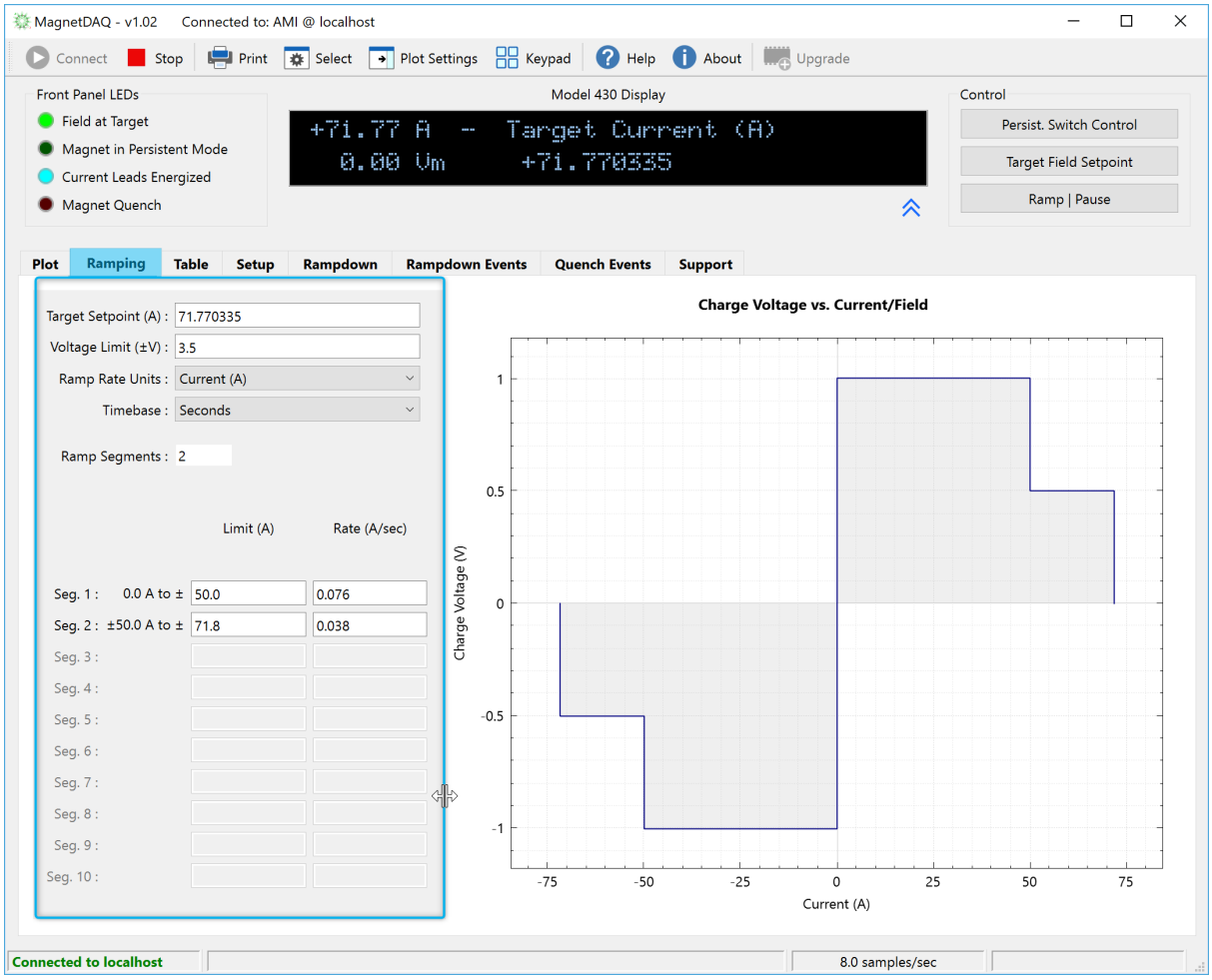


Step 4: Use the remote Keypad and/or the Ramping view to control the Model 430

Use the Keypad panel (bring it front using the "Keypad" toolbar button) to remotely control the target Model 430 just as if you are using the front panel keypad.



Use the Ramping tab view to enter the Target Setpoint, Voltage Limit, and Ramp Segments.

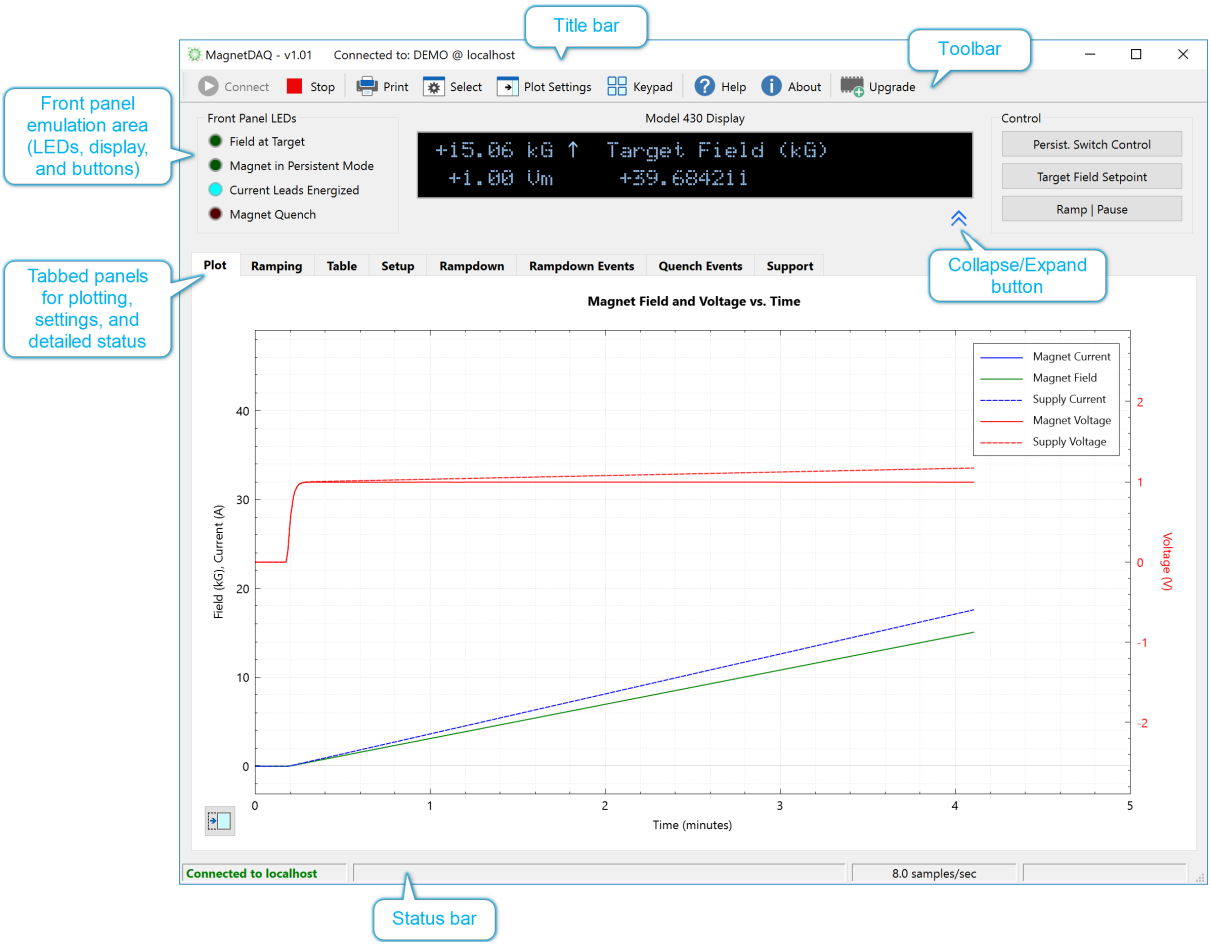


Feature Descriptions

2 Feature Descriptions

2.1 Main Window Layout

The Main Window of the Magnet-DAQ application consists of a title bar, a toolbar, a display/keypad emulation area, tabbed panels, and a status area.



Title bar

The title bar shows the application name and version, along with the present connection status including the IP name and IP address of the target Model 430. If the application is being used in a "slave" mode for multi-axis operation, the axis designation may also be displayed.

Toolbar

The toolbar contains common actions such as connection control, printing, configuration panel display, and help. See the Toolbar Items topic for more details.

Front panel emulation area

The front panel emulation area contains LED indicators, the emulated Model 430 display, and button controls for **Persistent Switch Control**, **Target Field Setpoint** entry, and **RAMP/PAUSE** control. These buttons function identically to those on the front panel of the Model 430.

Collapse/Expand button

The collapse/expand button toggles visibility of the plotting area and docked panels. The collapsed window can be resized and positioned and will remember the size and location separate from the expanded view. To show the plotting area and docked panels from the collapsed state, click the button again.

Tabbed views

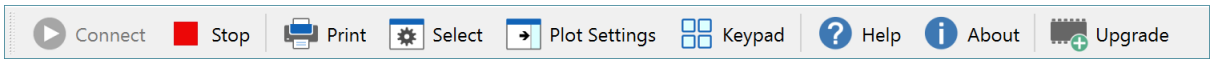
The tabbed area of the main window includes several views:

Plot	A graphical view of variables selected in the Plot Settings configuration panel vs. time.
Ramping	Definition of Target Field/Current, Voltage Limit, and Ramp Segments with charge rate graph of up to 10 ramp rate segments.
Table	A tabular view of target fields or currents with the ability to ramp to each automatically as well as optionally execute an application or script at each target.
Setup	Contains sub-views for the Setup categories of the Model 430.
Rampdown	Definition and a discharge rate graph of up to 10 rampdown segments.
Rampdown Events	Selectable list of rampdown events.
Quench Events	Selectable list of quench events.
Support	A view of all Model 430 settings and an option to compose support emails.

Status bar

The Status bar contains four sub-panels. From left-to-right, the panels include the connection status, general error and status messages, present sampling rate of the magnet state variables, and error notification.

2.2 Toolbar Items



The draggable toolbar, that is by default located at the top of the main window, contains the following items:

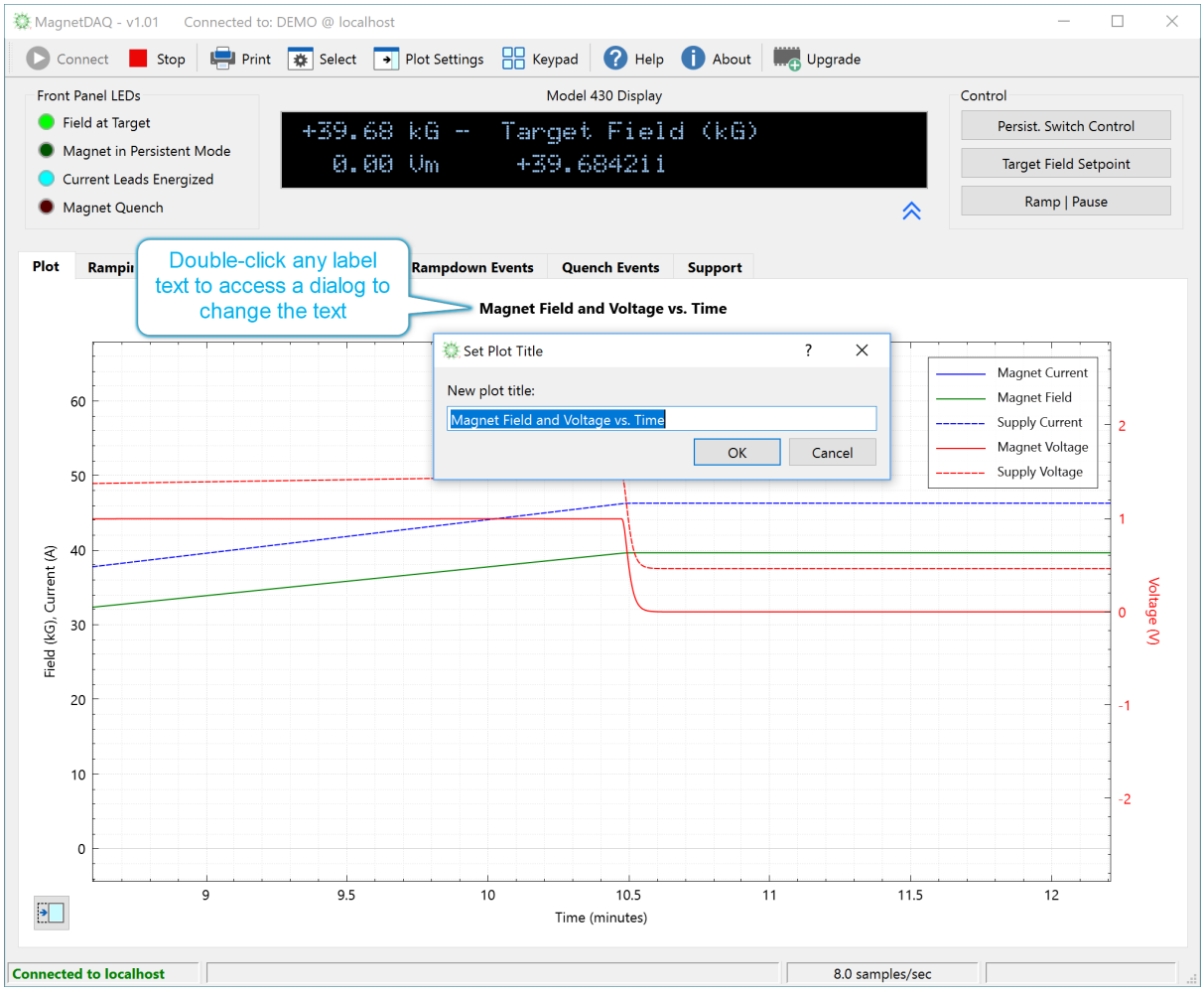
Connect	Initiates a connection to the specified IP address.
Stop	Closes a connection and stops all communication.
Print	Prints the currently displayed graph.
Select	Brings the Select device panel to front.
Plot Settings	Brings the Plots Settings configuration panel to front.
Keypad	Brings the emulated Keypad panel to front.
Help	Displays this Help file.
About	Displays information about the application and Qt version.
Upgrade	Displays the Firmware Upgrade Wizard on demand.

The toolbar can be dragged using the perforation at the left edge of the toolbar area and then docked to any edge of the main window.

2.3 Plot Tab

The Plot Tab provides a graphical view of variables selected in the Plot Settings configuration panel vs. time. The Plot Settings configuration panel provides the interface for selecting which variables are plotted.

The plot axes titles and plot line labels can be customized per your preference by simply double-clicking the axis title (or legend text) to access a dialog where the text can be changed.



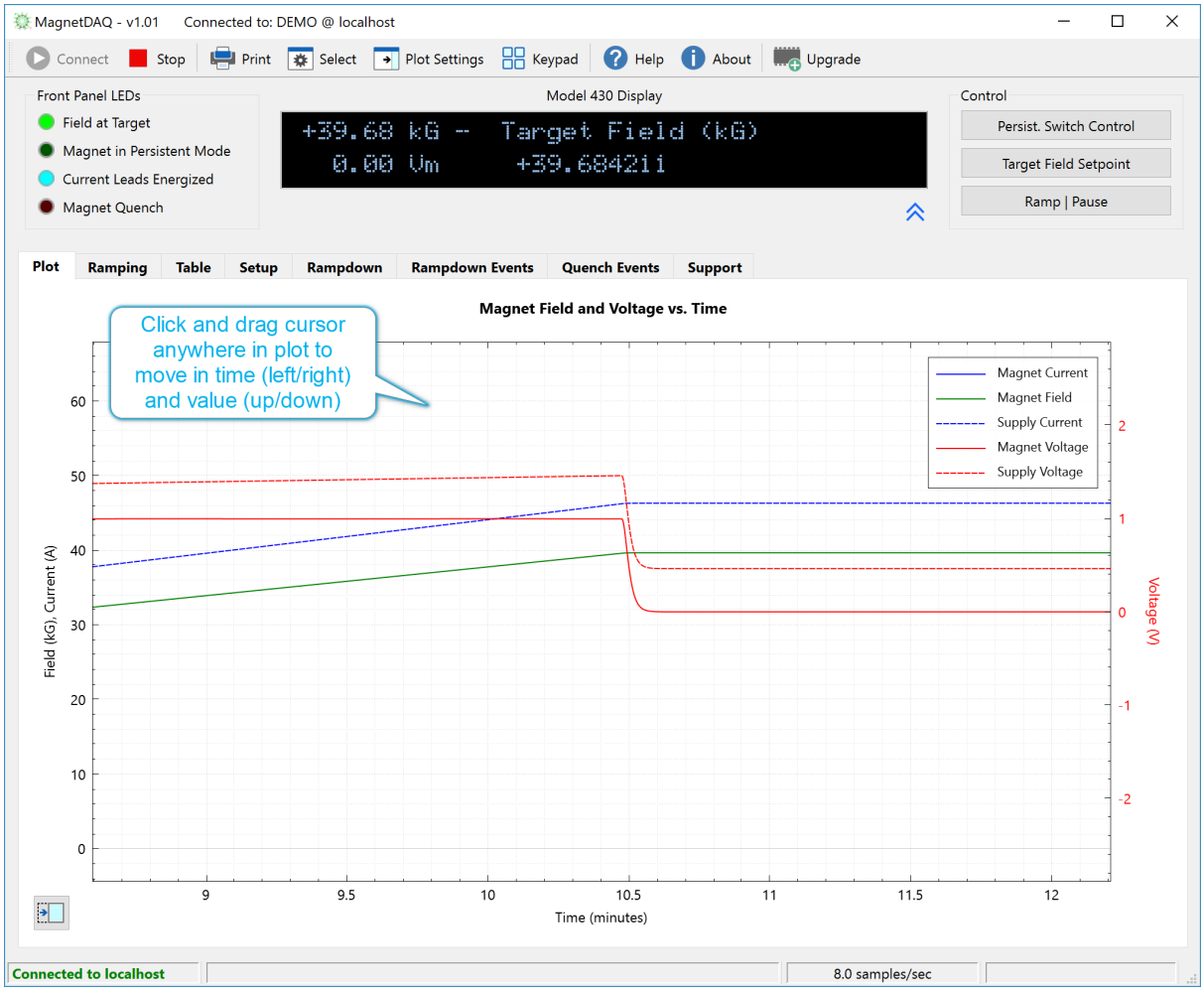
The default time base is minutes, but this can be changed in the Plot Settings configuration panel. The time base selection cannot be changed while connected.

Note that the plot has a bottom axis for the time, a left axis for the current/field values, and a right axis for voltage values. These axes can all be positioned and scaled independently by selecting each and clicking and dragging the cursor or using the scroll wheel (or the equivalent actions for a trackpad). The available actions are described below.

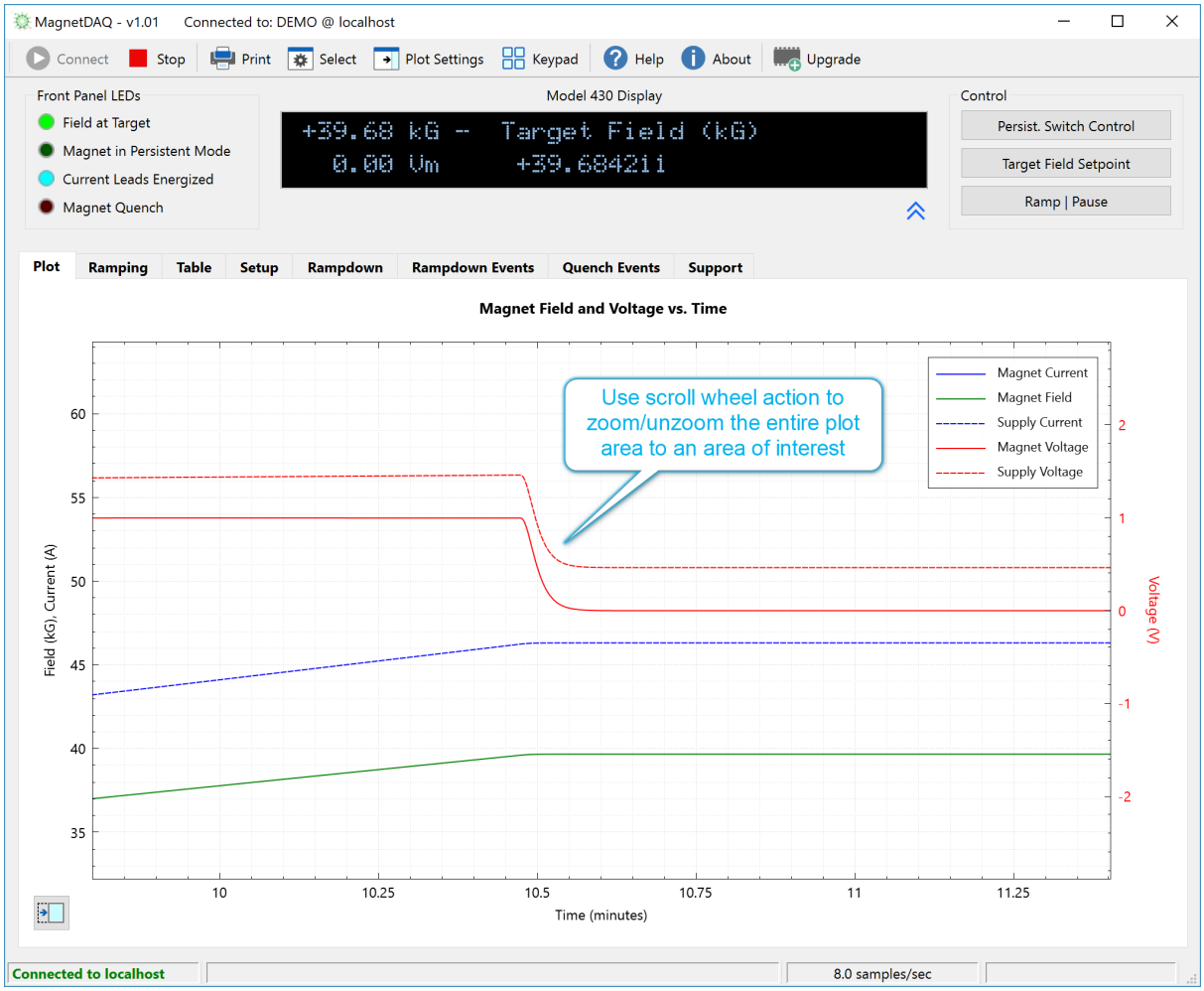
Plot Positioning and Scaling Actions

Click (or tap) in any white space to drag, or zoom/unzoom with scroll wheel

A click and drag action in any white space in the plot, with no axes highlighted, will position the time and left (current/field) axes simultaneously. Use the scroll wheel (or equivalent trackpad action) to simultaneously zoom/unzoom the plot area for the time and left axes. This combination of actions is a quick method for centering and zooming in on an a particular event to show more detail.

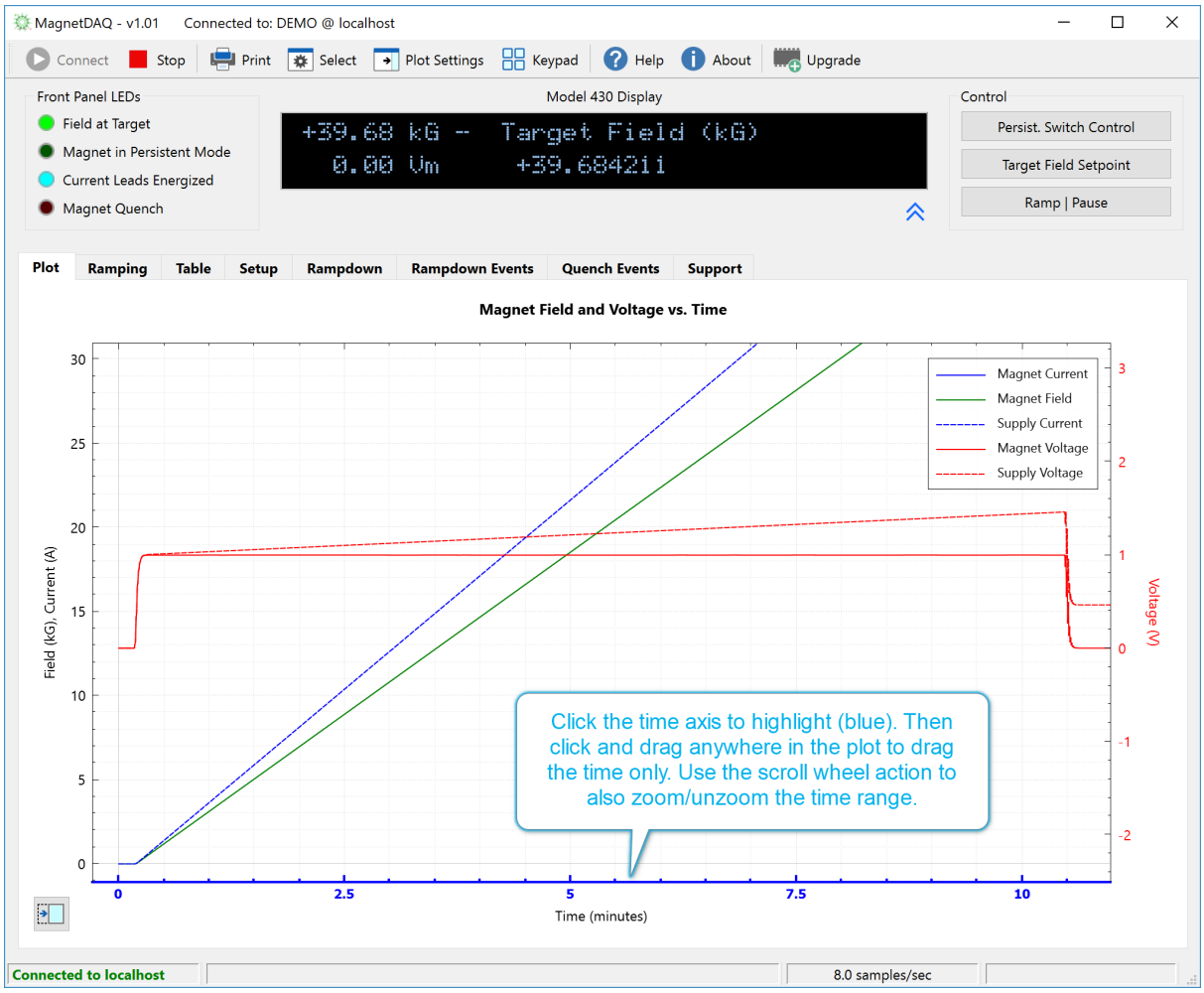


NOTE: With all zoom/unzoom actions, the position of the cursor determines the center point of the scaling action. The zoom/unzoom will hold the plot area at the cursor and scale to the left and right (or above and below) of the cursor position.



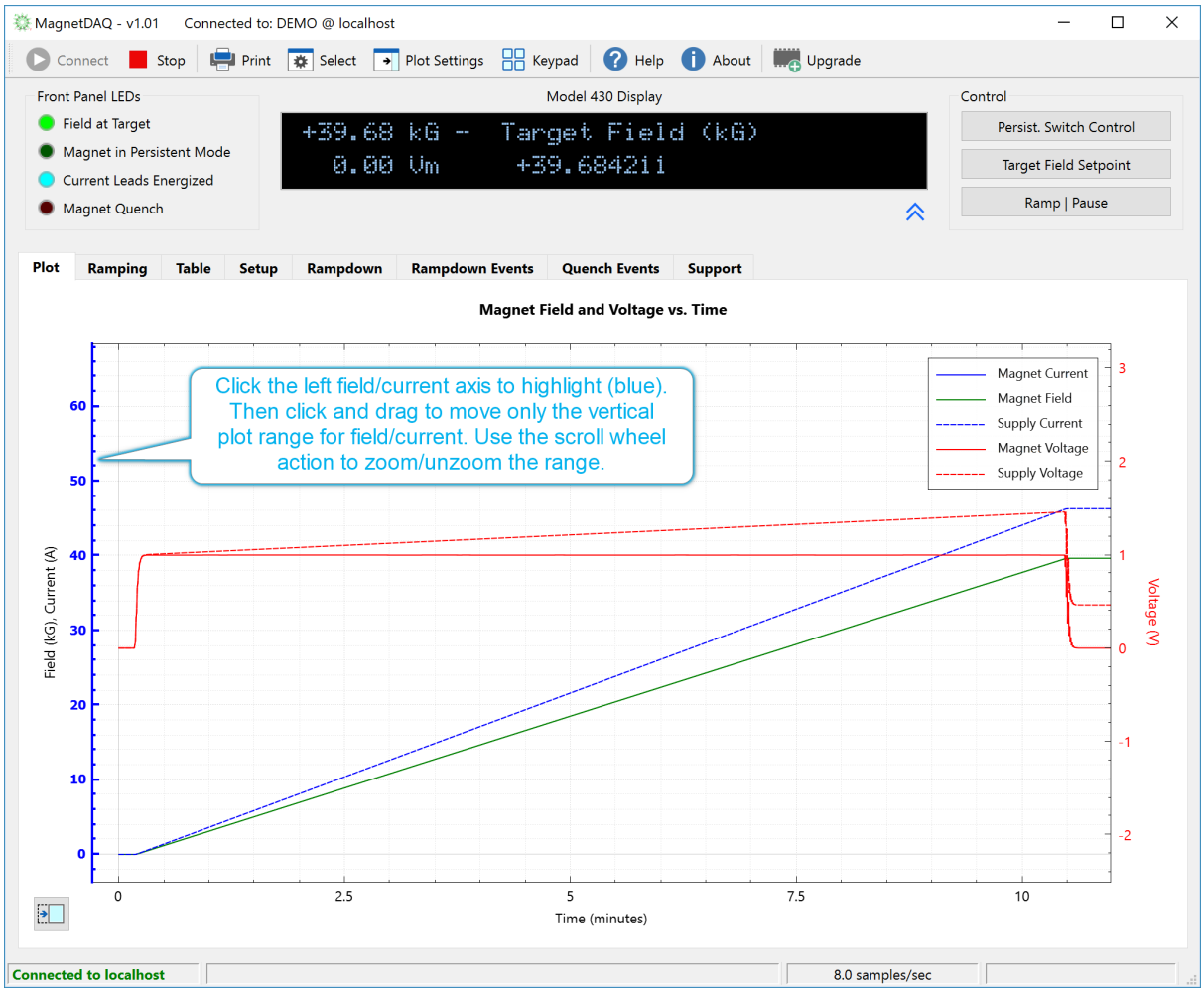
Select the time axis and then click and drag, or zoom/unzoom with scroll wheel

Select the time axis (it will highlight in blue) and then click and drag in any white space to independently position the time axis. While the time axis is selected, use the scroll wheel to independently zoom/unzoom the time axis.



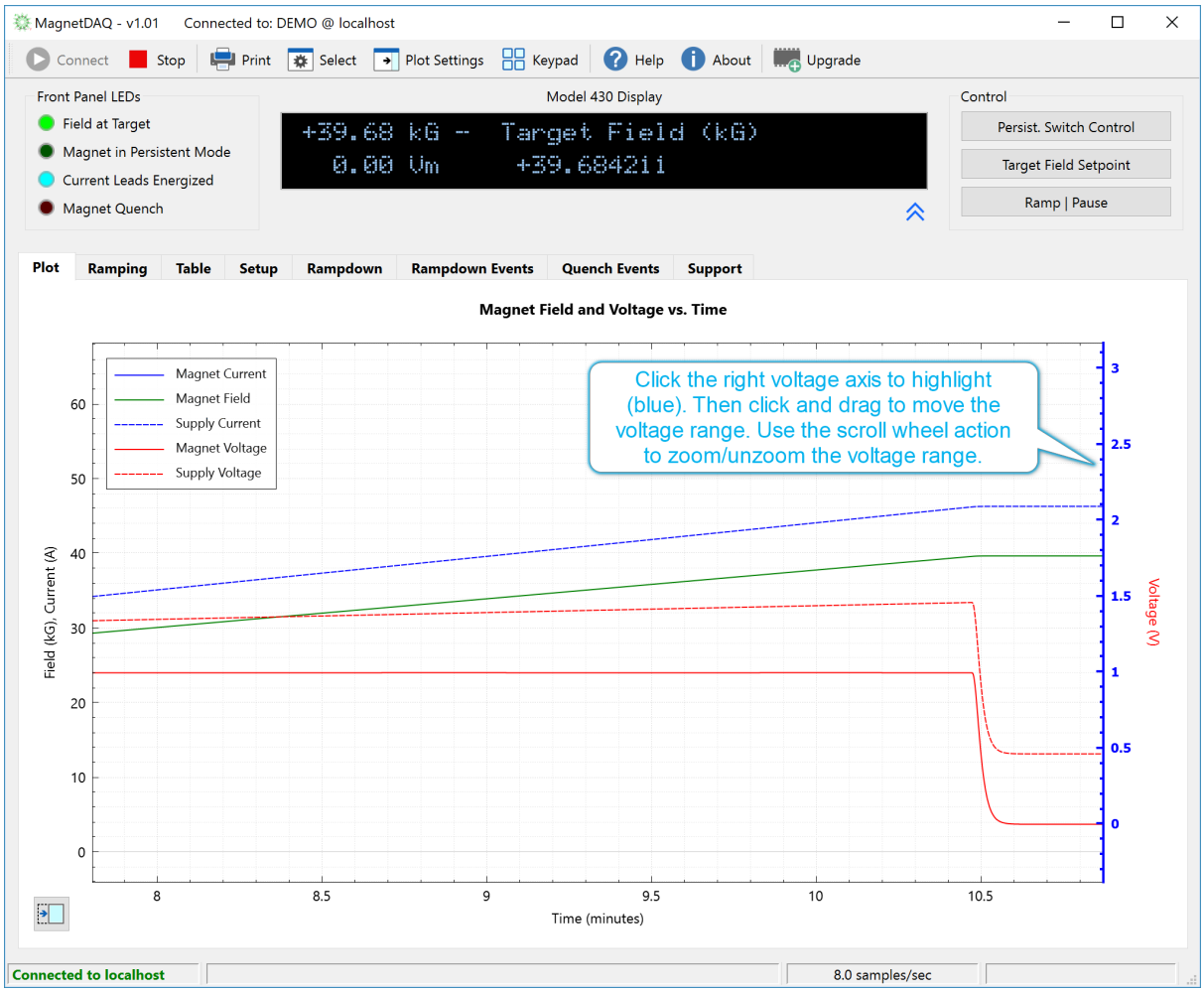
Select the left axis to drag, or zoom/unzoom, the current/field range

Select the left axis and then click and drag in any white space to independently position the current/field axis. While the left axis is selected, use the scroll wheel to independently zoom/unzoom the current/field axis.



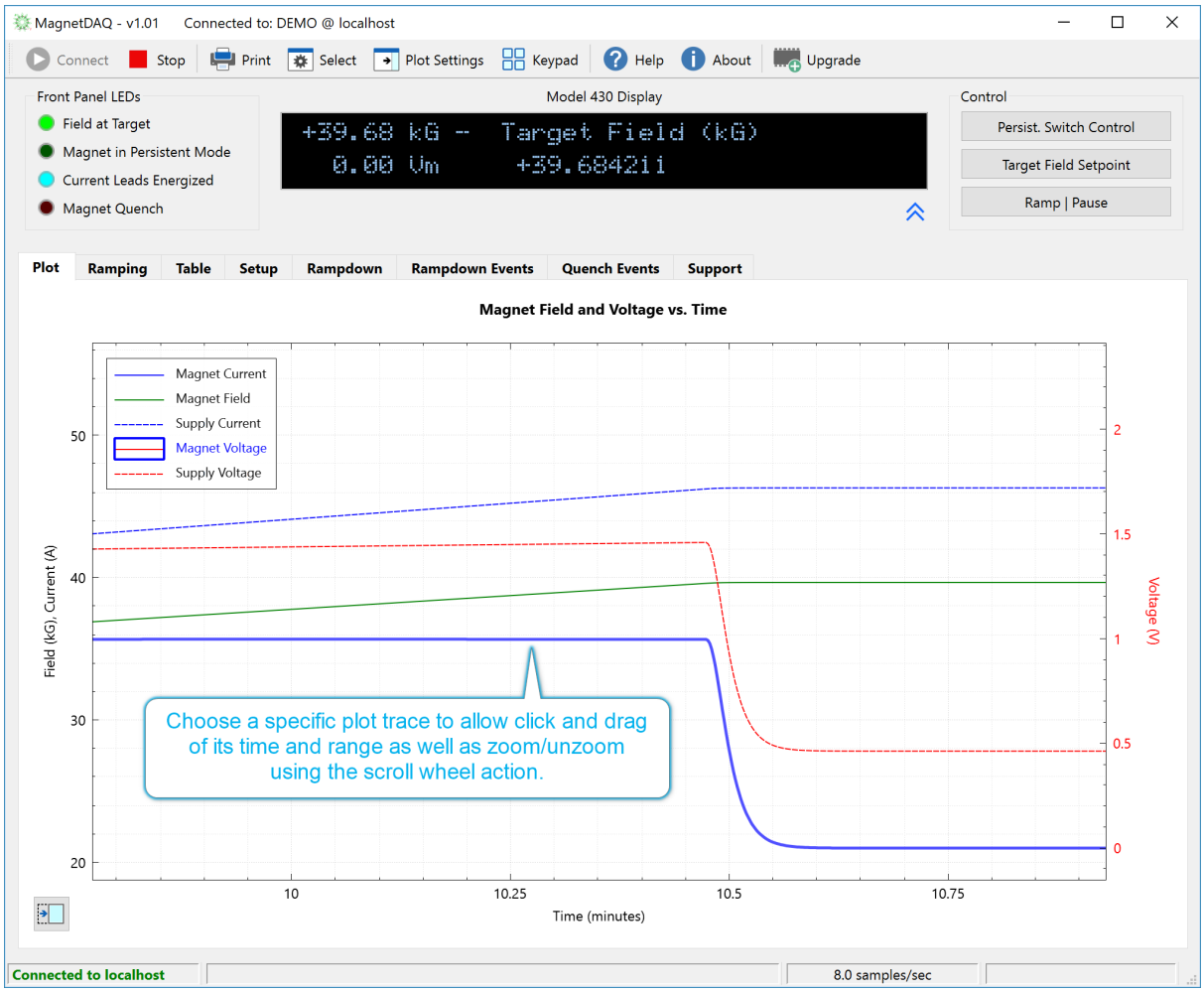
Select the right axis to drag, or zoom/unzoom, the voltage range

Select the right axis and then click and drag in any white space to independently position the voltage axis. While the right axis is selected, use the scroll wheel to independently zoom/unzoom the voltage axis.



Select a particular plot line to position or zoom/unzoom

Directly select any plot line (either in the plot area or in the legend) and click and drag in any white space to position the plot line. Use the scroll wheel to simultaneously zoom/unzoom the time and value axis for the selected plot line.



2.4 Ramping Tab

The Ramping Tab view provides the interface for setting the Target Setpoint, VoltageLimit, and up to ten (10) ramp segments for the target magnet. The ramp segments are bipolar, and the ramping magnitude is the same when charging (increasing current magnitude) and discharging (decreasing current magnitude).

The **Target Setpoint** field may be set to either amperes *or* field units of kilogauss or tesla. This is the target value for the automatic ramping mode.

The **Voltage Limit** field sets the bipolar output voltage limit that is applied to the *power supply output*. The Voltage Limit is an additional safety factor during charge/discharge to prevent accidental change of the magnet field at a higher rate than recommended in the magnet specifications. The Voltage Limit should include some overhead above the charge voltage to account for power lead losses (typically 10 to 20 milliohms x maximum current limit).

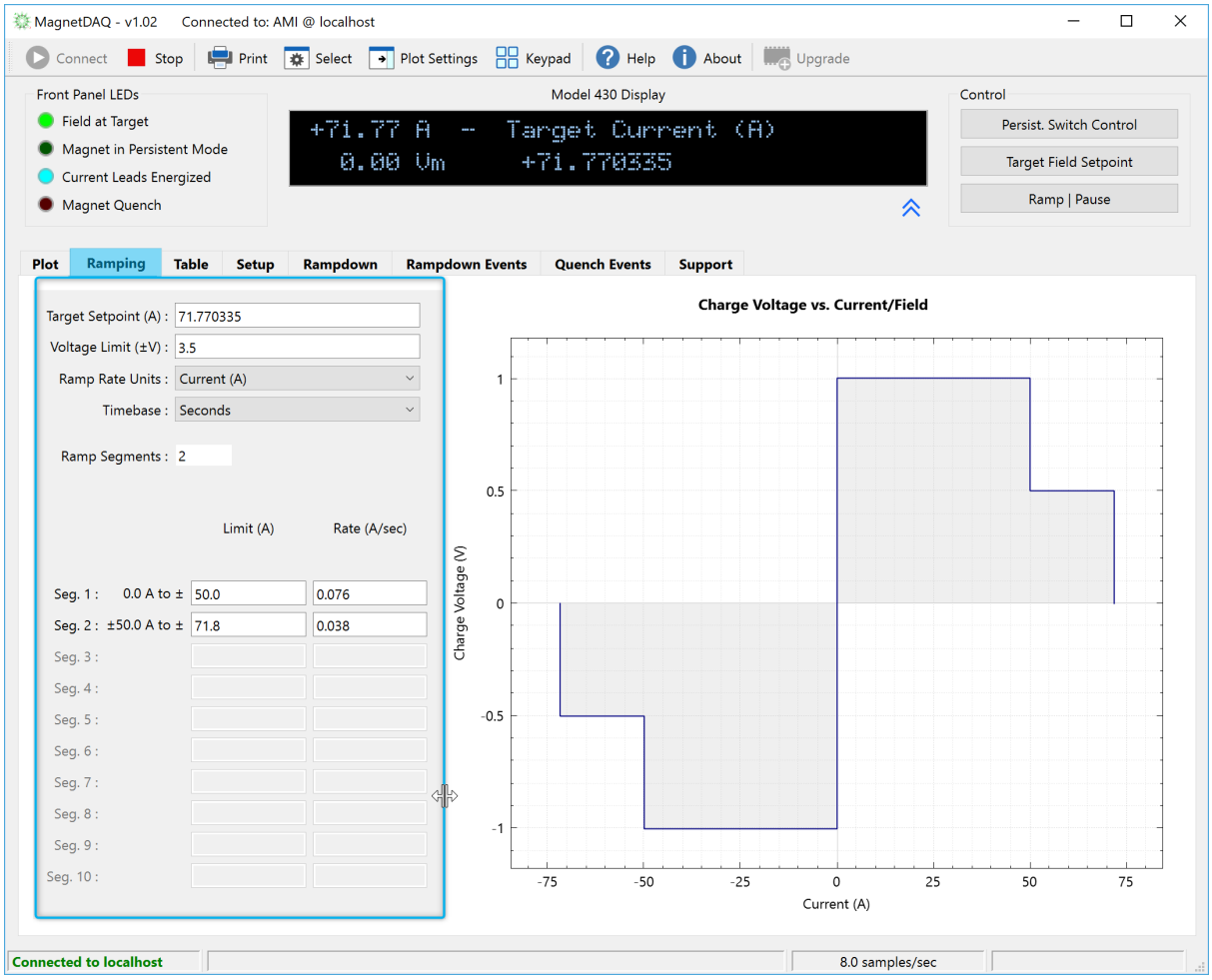
The **Ramp Rate Units** allows the user to choose to display the ramping parameters in amperes *or* field units of kilogauss or tesla. The *field* unit choice is controlled by the Model 430 setting (via the **Setup > Misc > Field Units** setting of the Model 430).

The **Timebase** of the ramp rate may also be set to either seconds or minutes. Minutes are recommended primarily for very low ramp rates as it provides for additional entry precision.

The **Ramp Segments** field allows entry of 1 to 10 ramp segments. As the ramp segments are adjusted, the entry fields for each segment are enabled and disabled as appropriate.

The plot at right in the view shows the charge voltage across the actual magnet for a given Magnet Inductance (see the **Setup > Load** screen). If the Magnet Inductance is not known and/or entered (i.e. zero), then the charge voltage graph is not drawn.

NOTE: If there are *several* ramping segments, the interface may appear more sluggish as the Magnet-DAQ application attempts to keep both the Model 430 and the displayed interface in sync. This is an unfortunate artifact of the underlying network-based communication.



NOTE: The view is split and the space allocated to the settings at left, and the plot at right, can be adjusted by clicking and dragging the split.

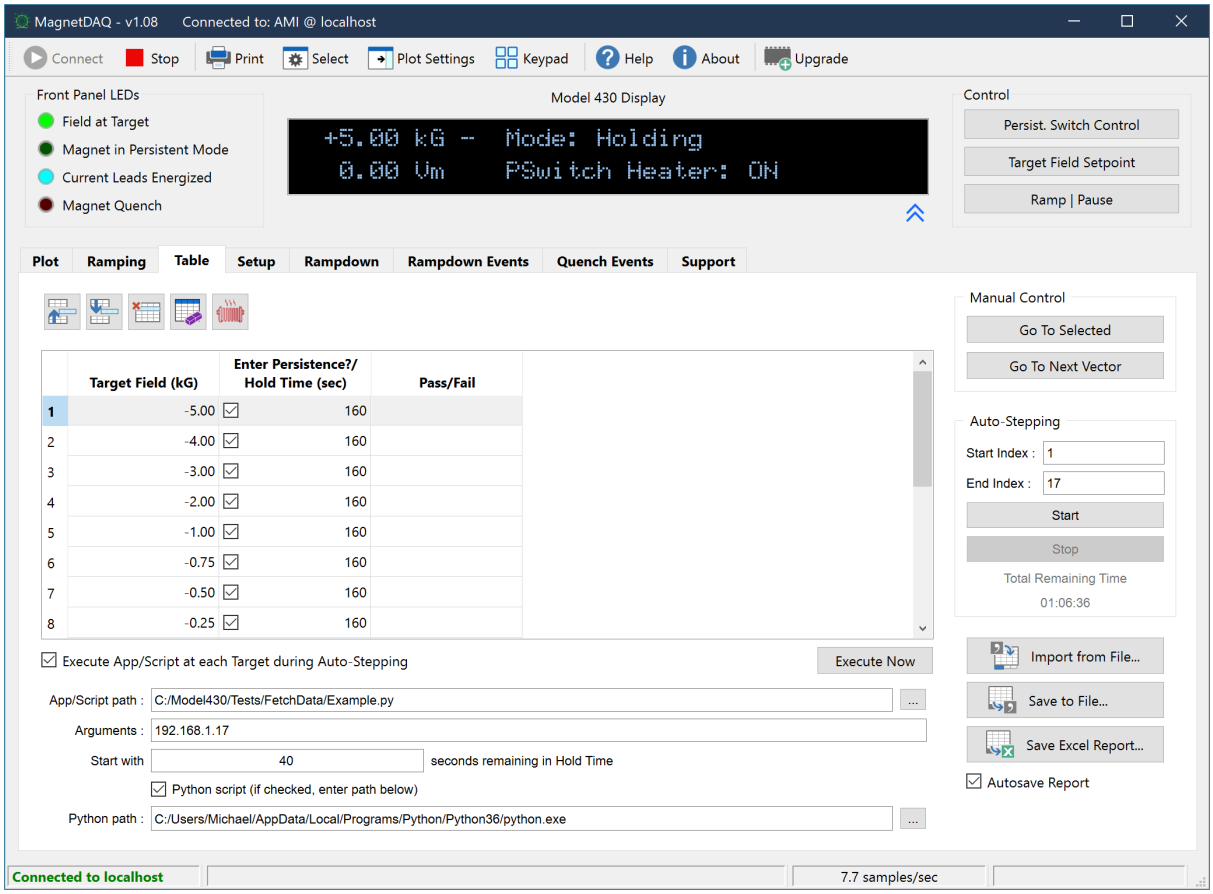
2.5 Table Tab

The Table tab provides an interface for loading, editing, and saving a list of field (or current) targets. Each target can be manually chosen, or the Autostep feature will visit each in order with a hold time as specified in the table.

An interface is also provided to specify an application or Python script that will run at a user-specified remaining time in the hold period at each target. This allows the user to wait for stabilization or other considerations before executing external data collection. The option to execute user code at each target, along with the Autostep feature, is a powerful combination. This means the user can concentrate efforts on developing specific data collection, or other tasks, for an experiment and allow Magnet-DAQ to handle the tedious job of managing the magnet state and field sequencing.

NOTE: This application provides basic table editing. Double-click a value in the table to edit its value. Use the icons above the table (as illustrated below) to add a row above or below the selected row, delete a row, or clear the entire table, respectively. More advanced editing of the table is better performed using Microsoft Excel (or other compatible application) that can load, edit, and save comma-separated values (CSV) files.

Table edit icons



NOTE: If the Model 430 to which the application *is connected* is configured for a persistent switch, then each target row includes an **Enter Persistence?** checkbox. If the box is checked, then the Model 430 will enter persistent mode (by cooling the switch) during the **Hold Time**, and then exit persistence (by heating the switch) at that target field or current. Ensure the heating and cooling times for the persistent switch are set according to the magnet manufacturer's recommendations.

The "heater coil" tool button above the table will toggle the **Enter Persistence?** checkbox states for *all the rows* in the table.

The **Import from File...** button displays a file chooser dialog for selection of a CSV-formatted file to import into the table. All prior values are overwritten. See the Table Import Format topic in the Advanced Features section of this Help for details on the import format.

The **Save to File...** button will export the table contents to the selected file in CSV format.

The **Save Excel Report...** button exports the table contents to a formatted Excel file with additional information such as the ramp segments and date/time.

If **Autosave Report** is checked, the application will attempt to autosave the formatted Excel report when an Autostep session completes (either successfully or unsuccessfully) to the same folder from which the last **Import from File...** file was loaded. Multiple saves will contain an incremented integer value in the filename to avoid overwriting previous autosaves.

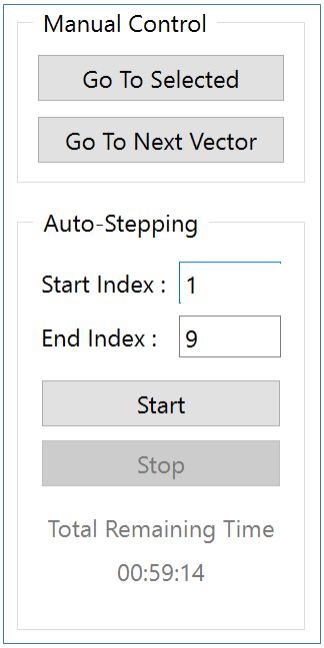
Two important terms are used in reference to the table: *Target* and *Auto-Stepping*

The **Target** is defined as the field that is presently programmed in the magnet system; or in other words, the Target Setpoint along with the ramp rate values for the Model 430 that have been programmed to realize the value. If a target reaches the **HOLDING** state, then it is marked as "Pass" in the Pass/Fail column.

NOTE: The table function observes *all* defined ramping segments in the Ramping tab to reach the desired field.

In addition to manual controls which will hold the Target indefinitely, the table also has an **Auto-Stepping** sequencing feature whereby the list of targets can be sequentially executed by the application. The "Hold Time (sec)" column in the vector table defines how long the Auto-Stepping feature will remain in the **HOLDING** state (i.e. a "dwell time") once the target is reached.

The right-side controls are described below:



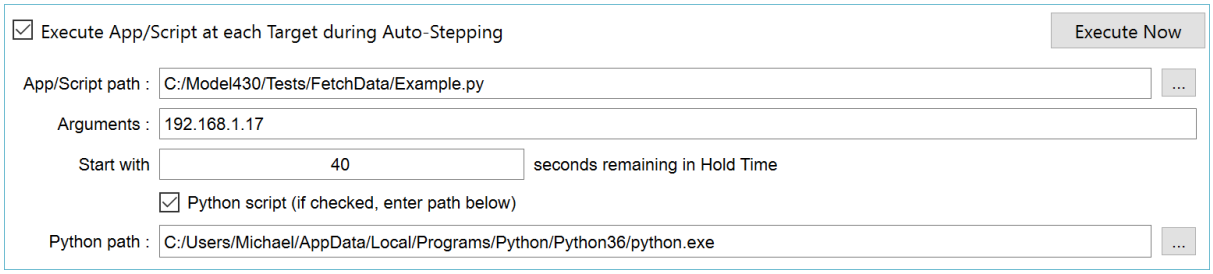
Go To Selected	Makes the presently selected row the Target and commands the Model 430 to ramp to the value.
Go To Next Vector	Makes the next row in the table the Target and commands the Model 430 to ramp to the value.
Start Index	Specifies the starting row index of the Auto-Stepping sequence.
End Index	Specifies the ending row index of the Auto-Stepping sequence.
Start and **Stop**	Starts and stops the Auto-Stepping function.
Total Remaining Time	Estimate of total time (hh:mm:ss) to complete the Auto-Stepping function including any switch heating/cooling times.

Execution of external application or Python script at each Target

If the **Execute App/Script** checkbox is checked, then a details footer appears with options to specify the location, arguments, and launch time within the Hold Time at the target. Ensure enough time is allowed to complete the app/script before the Hold Time expires and the target is changed. The ellipsis button at the end of the **App/Script path** field displays a chooser dialog.

The **Execute Now** button will execute the app/script on-demand (but is disabled when Autostep is active). Among other uses, this can be useful for debugging purposes.

If the **Python script** box is checked, then a text box will appear for specification of the path to the desired Python executable. This allows the user to select a specific Python version for execution rather than relying on the system path.



Example Entries for External Python Script Execution

In the above example, the Example.py script has been specified with a single argument which is the IP address of the digital multimeter device. The script will be executed with 40 seconds remaining in the Hold Time. The Python path field points to an installation of Python version 3.6.

The exact execution command sent to the operating system for this example would be:

C:/Users/Michael/AppData/Local/Programs/Python/Python36/python.exe C:/Model430/Tests/FetchData/Example.py 192.168.1.17

Note that the arguments are ordered for the Python script to consume, and not the Python executable. The Example.py script is provided in the Python Examples section of this Help.

Special variables available for use in the Arguments field

Special variables that pass the present magnet state and/or the target are available in the Arguments field. The following table describes the available variables which are replaced in the execution command sent to the operating system with the present value of each variable. Each command may have several synonyms and can recognize either values *enclosed* in % (Windows shell variables style) or *prefixed* with the $ character (Unix shell variables style). This allows external scripts or executables to access the present field or target field for purposes such as building DAQ tables.

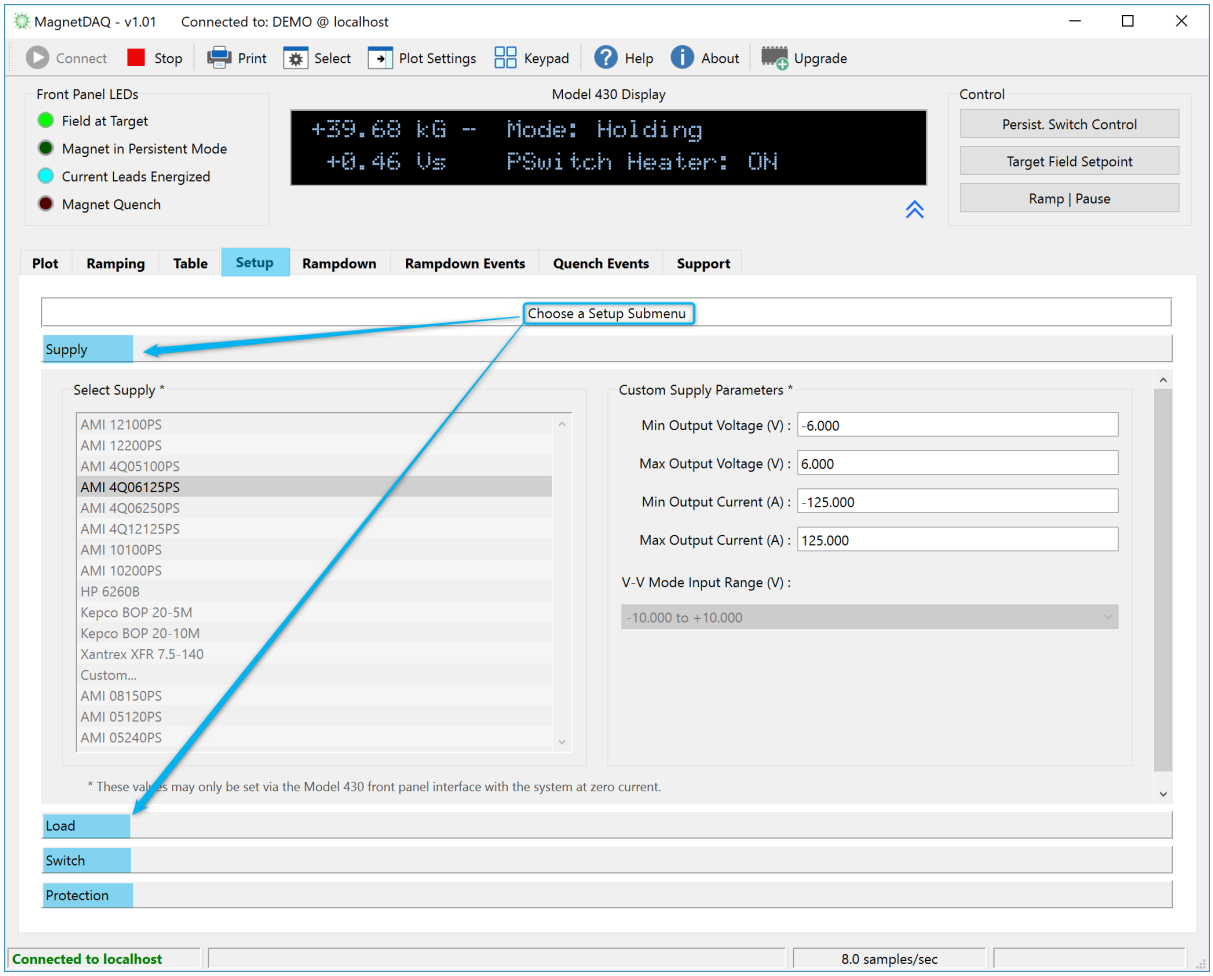
The RampToZero.py script is provided in the Python Examples section of this Help illustrating the use of the special variables as Arguments.

Variable Synonyms	Description
%IPADDR% **$IPADDR**	Inserts the IP address of the presently connected Model 430 device.
%CURR:MAG% **$CURR:MAG**	Inserts the present magnet current in amperes.

%CURR:REF% **$CURR:REF**	Inserts the noiseless 430-internal digital ramp reference in amperes.
%FIELD:MAG% **$FIELD:MAG**	Inserts the present magnet field in the selected field units. The present magnet field and the target magnet field should match when the HOLDING state is reached.
%TARG:CURR% **$TARG:CURR**	Inserts the target setpoint magnet current in amperes. The present magnet current and the target magnet current should match when the HOLDING state is reached.
%TARG:FIELD% **$TARG:FIELD**	Inserts the target setpoint magnet field in the selected field units. The present magnet field and the target magnet field should match when the HOLDING state is reached.

2.6 Setup Tab

The Setup Tab provides a multi-paged interface for viewing and setting numerous parameters for the target Model 430. The view is contains a stacked view with Setup Submenus for the **Supply**, **Load**, **Switch**, and **Protection** settings. These submenus correspond to those of the Model 430 accessible via the front panel keypad. Simply click on the page name to access each.



Supply page

The Supply settings are *read-only* to prevent accidental changes. An incorrect change in the supply setup parameters could easily trigger a magnet quench if not performed with an abundance of caution. Therefore, a design choice was made to not allow remote adjustment of the supply setup parameters at this time.

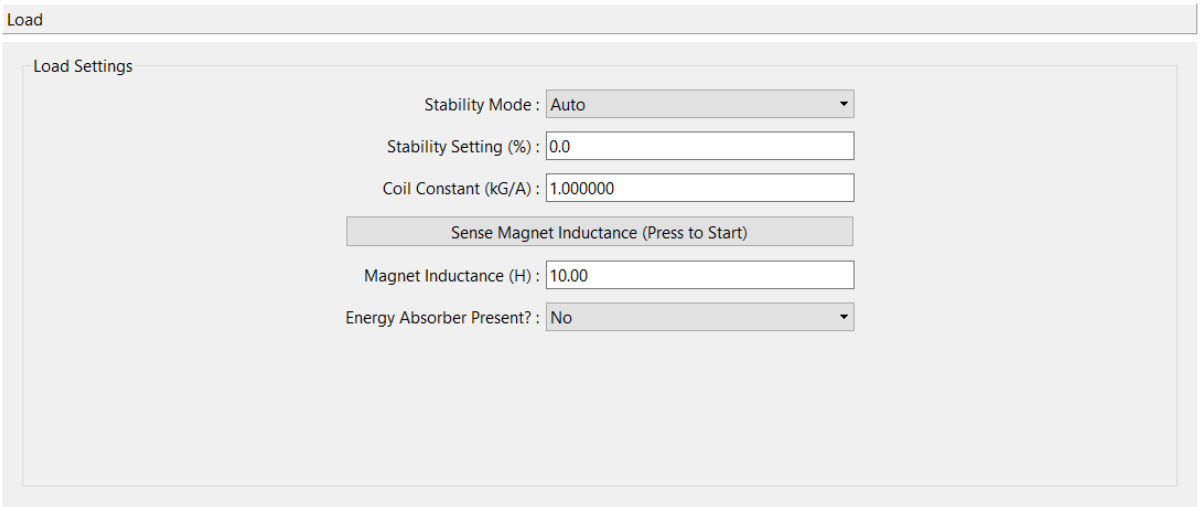
These correspond to the identical settings available in the **Setup > Supply** menu accessible from the Model 430 front panel. Refer to the Model 430 manual for a detailed explanation of each parameter.

Load page

The Load settings include the **Stability Mode**, **Stability Setting**, **Coil Constant**, **Magnet Inductance**, and whether an energy absorber (typically a Model 601) is present and connected to the magnet circuit. These correspond to the identical settings available in the **Setup > Load**

menu accessible from the Model 430 front panel. Refer to the Model 430 manual for a detailed explanation of each parameter.

The Coil Constant units will change (kG/A or T/A) with the selected field units of the Model 430.

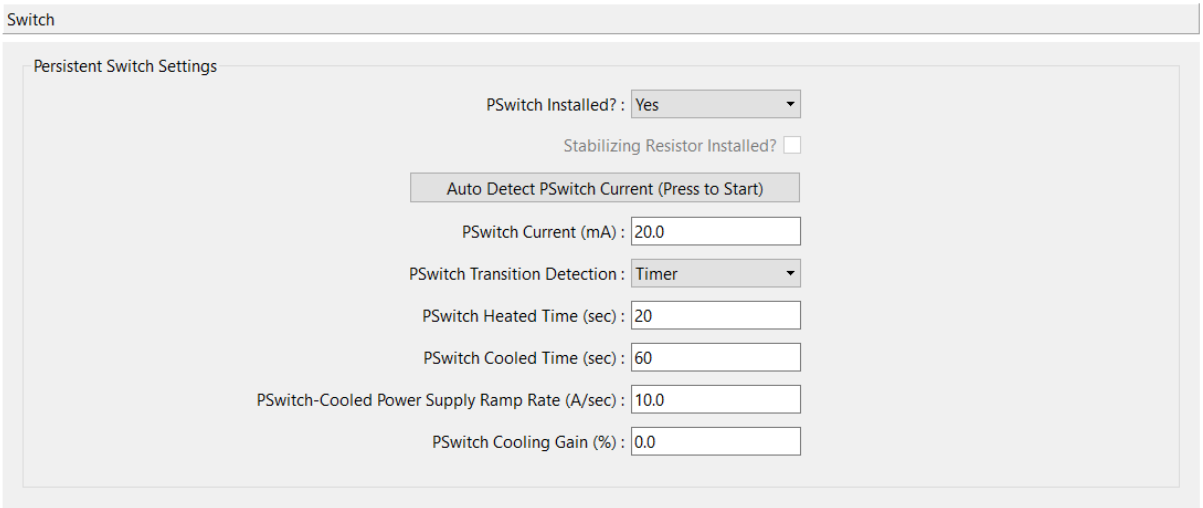


NOTE: The **Sense Magnet Inductance** button will initiate a magnet inductance measurement via the Model 430. Please note that the initiating the sensing function using this button will block all remote I/O with the Model 430 until the inductance sensing process completes. An alternative, non-blocking action is to use the Keypad emulation to initiate the same function via the instrument display.

Switch page

The Switch settings include the whether a switch is installed, **PSwitch Current**, **PSwitch Transition Detection** type, **PSwitch Heated Time**, **PSwitch Cooled Time**, **PSwitch-Cooled Power Supply Ramp Rate**, and **PSwitch Cooling Gain**. These correspond to the identical settings available in the **Setup > Switch** menu accessible from the Model 430 front panel. Refer to the Model 430 manual for a detailed explanation of each parameter.

If a switch is not installed, an optional stabilizing resistor indication can be selected if a stabilizing resistor is present in the magnet circuit.



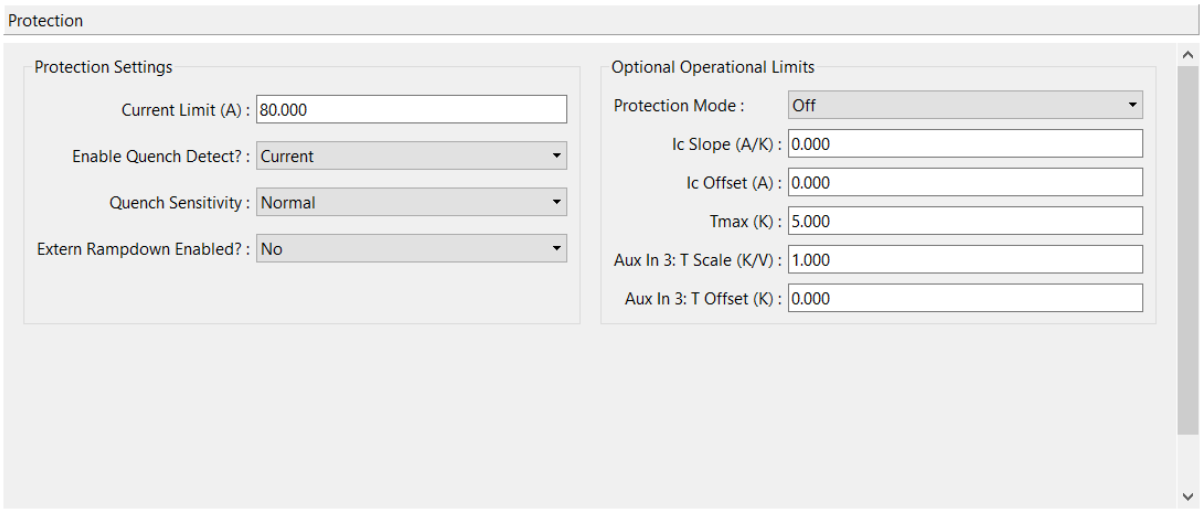
NOTE: The **Auto Detect PSwitch Current** button will initiate the switch heater current detection function via the Model 430. Please note that initiating the sensing function using this button will block all remote I/O with the Model 430 until the sensing process completes. An alternative, non-blocking action is to use the Keypad emulation to initiate the same function via the instrument display.

Protection page

The Protection settings include **Current Limit**, quench detection type selection and enable, **Quench Sensitivity**, and external rampdown enable. These correspond to the identical settings available in the **Setup > Protection** menu accessible from the Model 430 front panel.

In addition, the optional Operational Limits section can be setup with all the associated parameters including **Ic Slope**, **Ic Offset**, **Tmax**, **T Scale**, and **T Offset**.

Refer to the Model 430 manual for a detailed explanation of each parameter.



2.7 Rampdown Tab

The Rampdown Tab provides an interface for viewing and setting optional ramp segments that are used in the event of an *external rampdown event*.

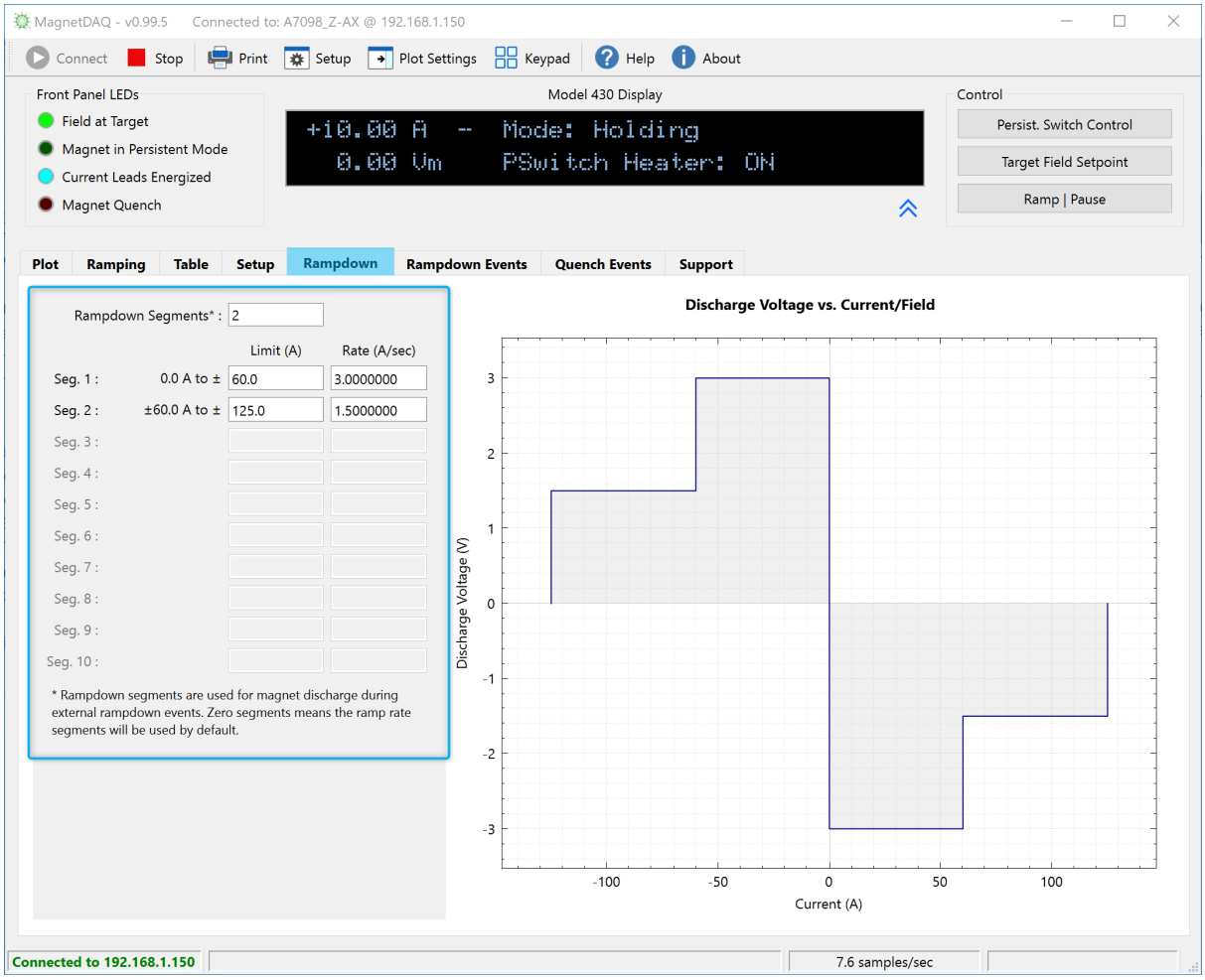
The external rampdown feature uses a rear panel input of the Model 430 (pins 6 and 7 of the Quench I/O connector). The external rampdown discharge ramp segments have no physical interface via the front panel of the Model 430. The rampdown segments may only be specified using the remote interfaces. The Rampdown Tab provides a convenient interface for specifying these segments.

The external rampdown feature is intended to provide a rapid and non-interruptable discharge of the magnet in the event of a situation where the magnet cannot continue operation such as loss of coolant, facility power loss, etc.

As is the case with the normal ramping function, the external rampdown function can accept up to 10 segments. If no segments are specified (i.e. 0 Rampdown Segments), then the values in the Ramping Tab are used during the rampdown process.

See the Model 430 manual for more detailed information regarding the external rampdown feature.

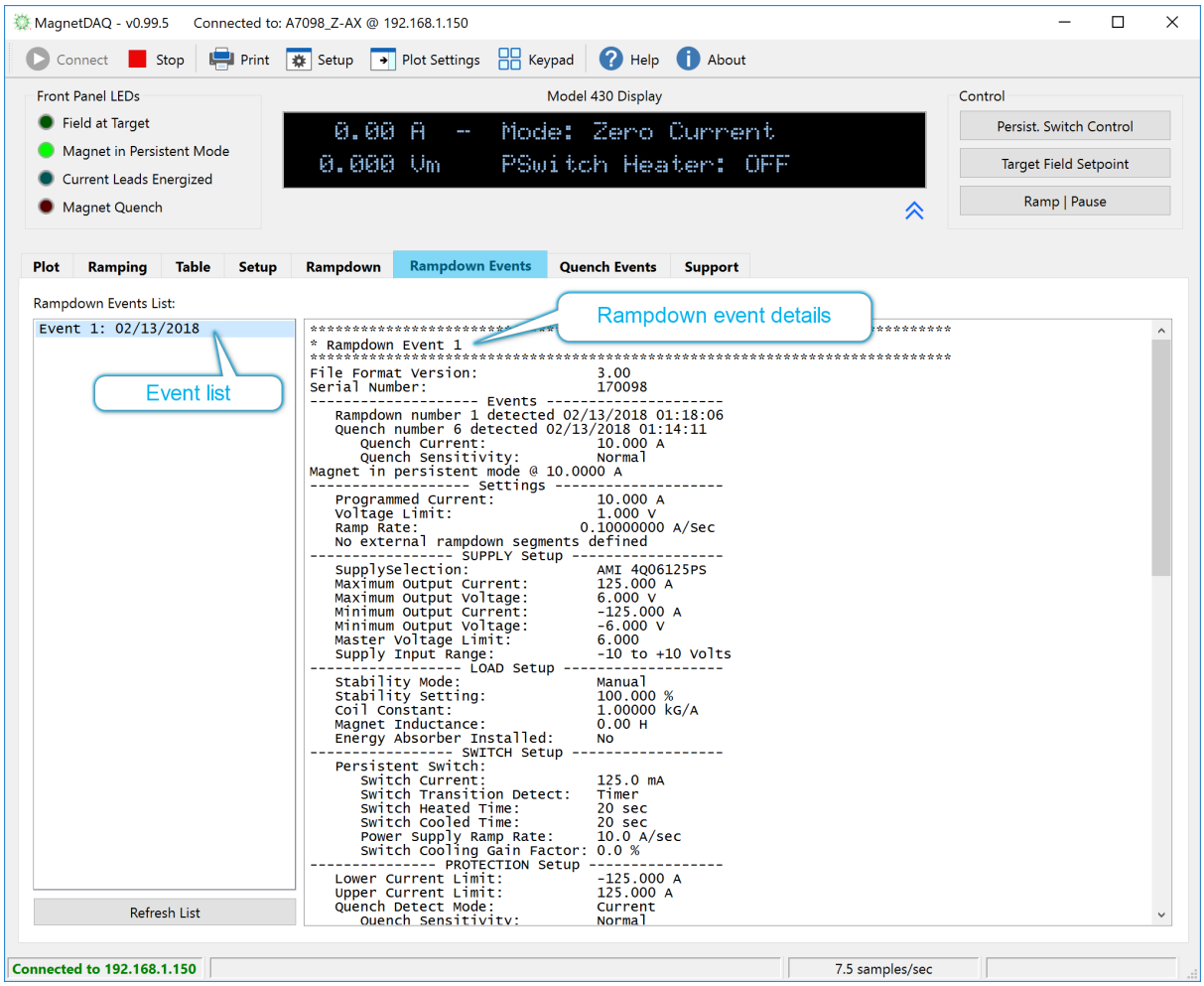
NOTE: The view is split and the space allocated to the settings at left, and the plot at right, can be adjusted by clicking and dragging the split.



2.8 Rampdown Events Tab

The Rampdown Events Tab provides a list of rampdown events recorded by the instrument with a timestamp of the initiation of each event. The complete state of the Model 430 is recorded at the initiation of each event and is provided in the text space to the right upon selection of an event from the list.

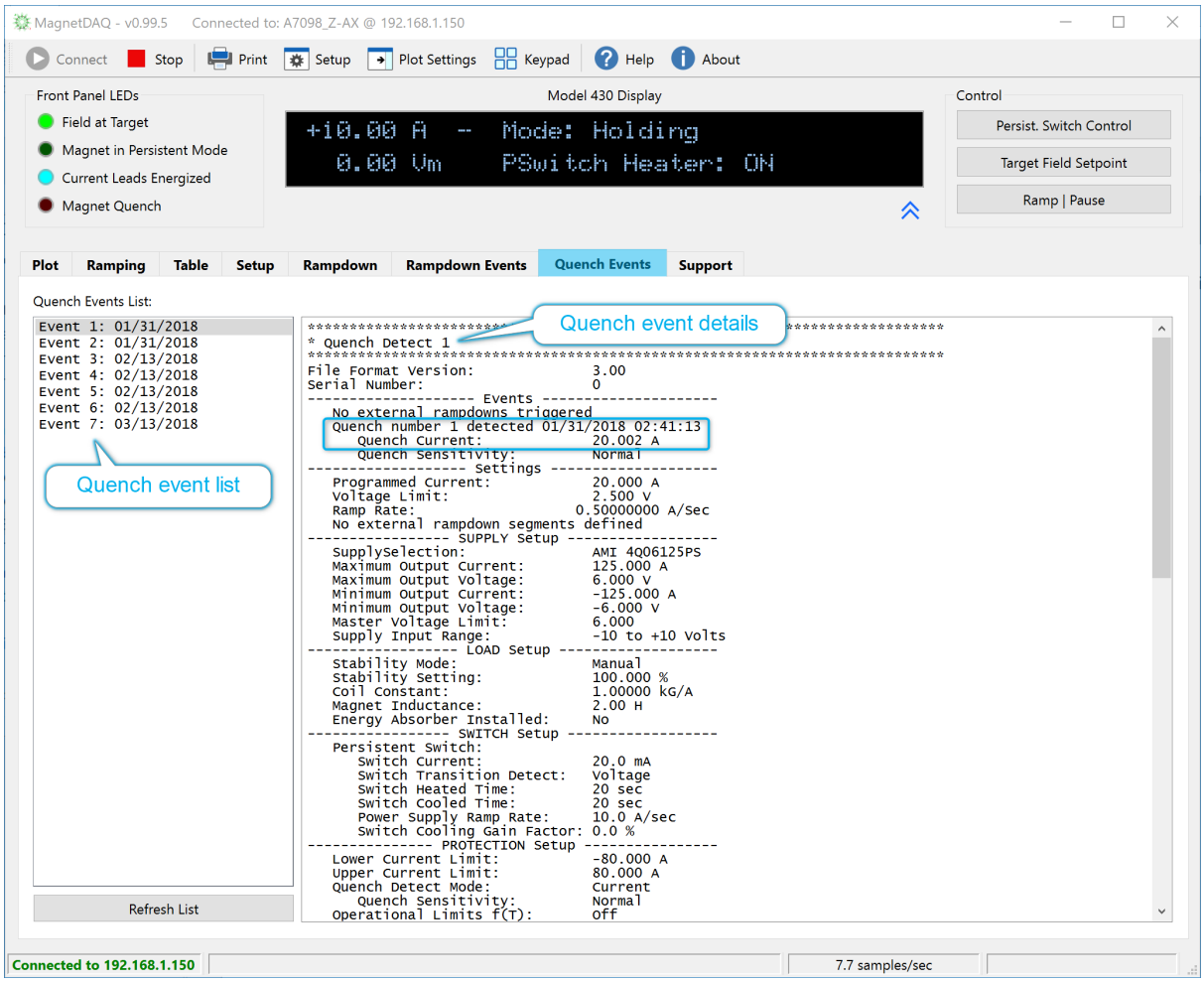
The event list is automatically refreshed each time the Rampdown Events Tab is entered.



2.9 Quench Events Tab

The Quench Events Tab provides a list of quench detection events recorded by the instrument with a timestamp of the initiation of each event. The complete state of the Model 430 is recorded at the initiation of each event and is provided in the text space to the right upon selection of an event from the list.

Up to the last 100 quench events will be fetched from the instrument. The event list is automatically refreshed each time the Quench Events Tab is entered.

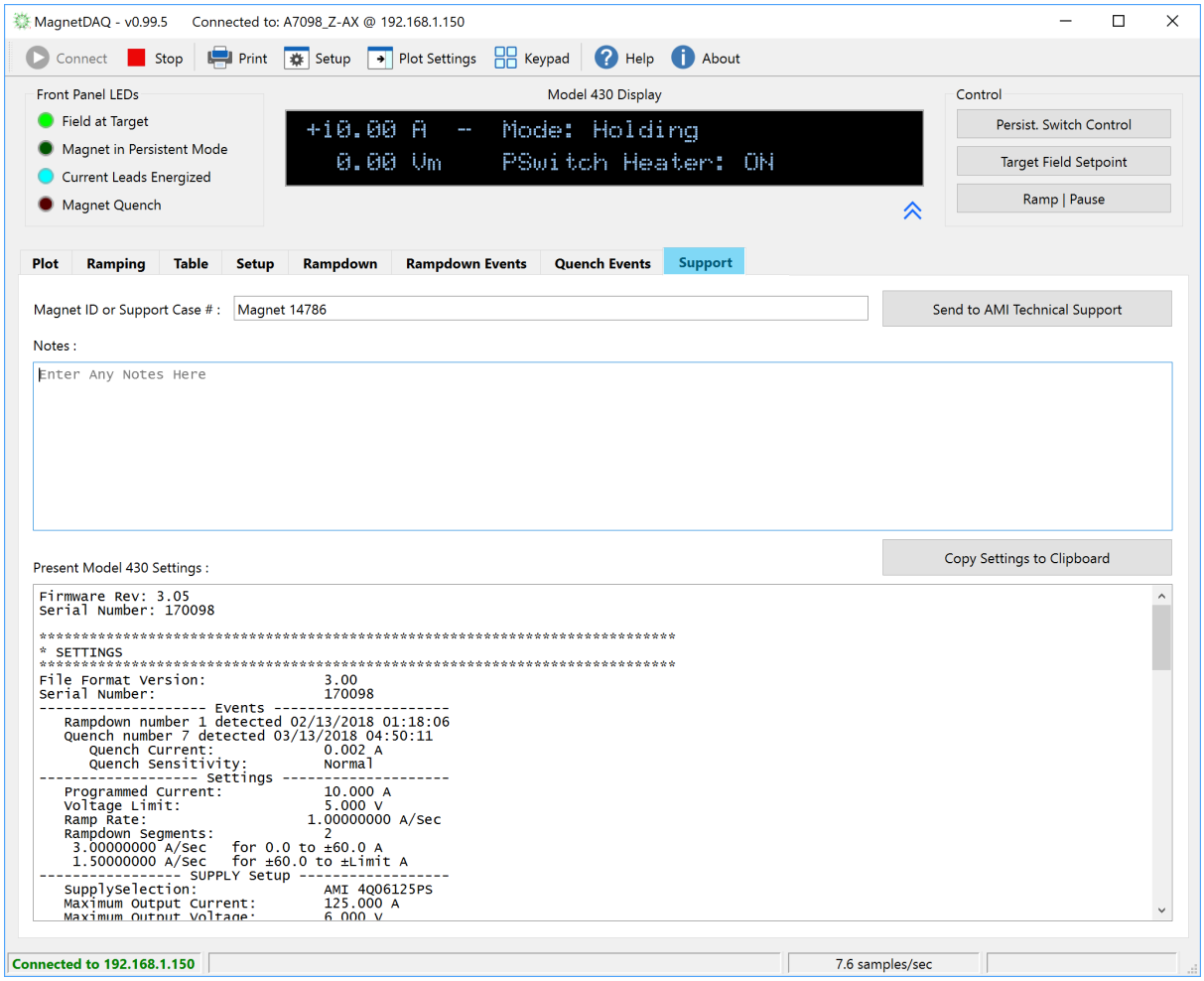


2.10 Support Tab

The Support Tab provides fields for contacting an Authorized AMI Technical Support Representative, including a Magnet ID or Support Case # field (the magnet number is very useful here), a Notes area for the problem description, and the complete settings of the Model 430.

You may choose the "Send to AMI Technical Support" button to compose an email in the default email application as configured on your computer, or you can copy the Model 430 settings to the clipboard and paste it into a message of your own composition.

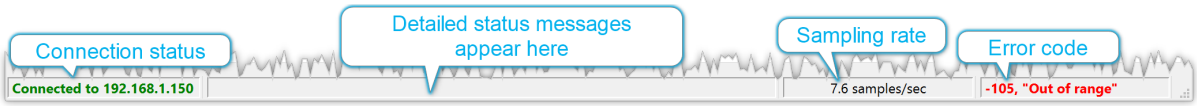
Providing this information to AMI Technical Support can serve to expedite a more detailed response.



2.11 Status Bar

The Status Bar at the bottom of the Main Window consists of four panes of information:

- The leftmost pane indicates the connection status of the remote instrument.
- The next pane to the right provides detailed status or error information. Critical error information will be shown as red text.
- The next pane to the right indicates the present sampling rate of the plotted data.
- The rightmost pane provides the error message codes and short error summary as returned by the instrument.
- Click the Error code pane to open an error history dialog for the present connection.

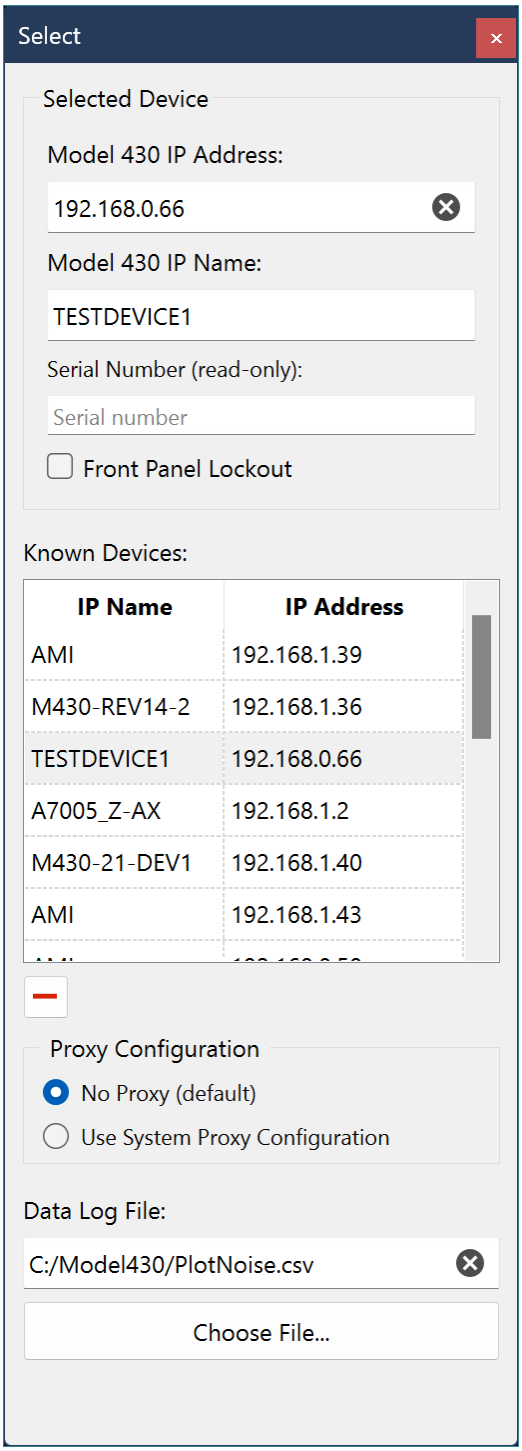


Error History Dialog

Configuration Panels

3 Configuration Panels

3.1 Select

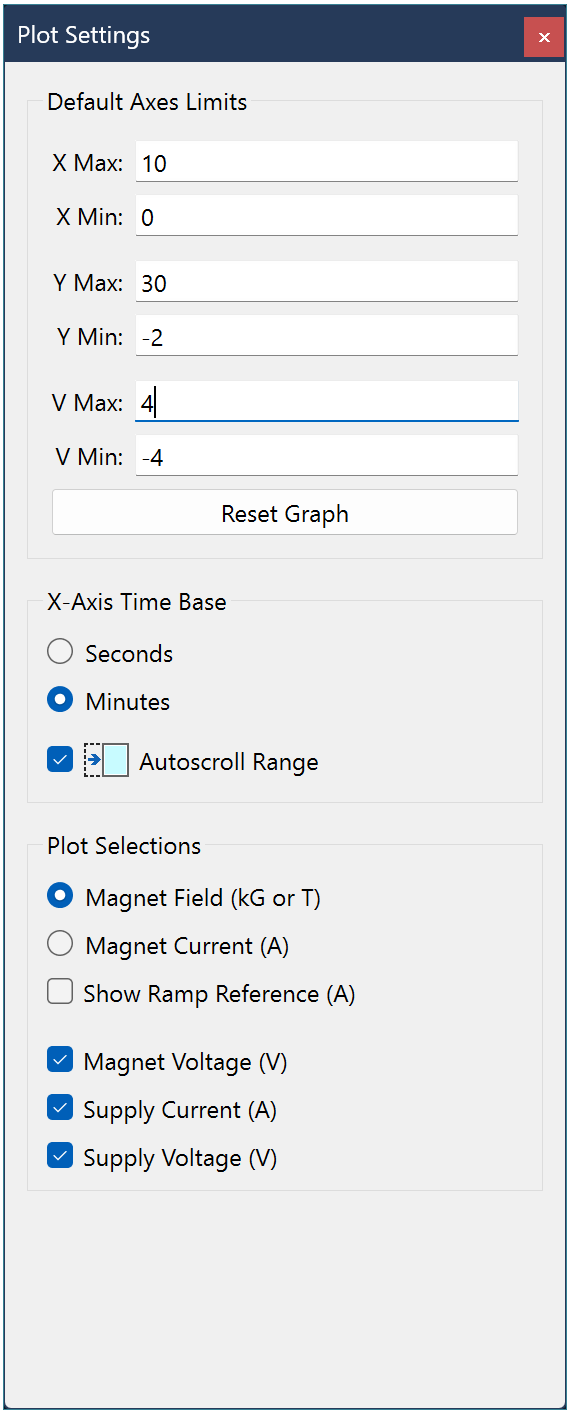


The Select panel is accessible via the Toolbar icon and is the means of selecting a target Model 430 for a connection. The panel can be docked to either the left or right side of the Main Window, or it can be dragged off the Main Window to "float" on the desktop. The fields and interface items are described below:

Model 430 IP Address	The IP address of the target instrument on the local network. This can be manually entered or can be auto-filled with the information from a selection from the "Known Devices" list (see below).
Model 430 IP Name	This is the textual name assigned to the instrument at the IP address. This is read on connection and can be edited once a connection is active.
Serial Number	Read-only display of the instrument's serial number.
Front Panel Lockout	When checked, the local keypad is made inactive on the front panel of the physical instrument. The remote keypad emulation remains functional. This can be used to prevent a user at the instrument from interfering with a remote control session.
Known Devices	A list of previously successfully connected instruments. Select from the list before making a connection to choose from the known devices. Use the button to delete selections from the

	table. Click on a heading to sort the list by name or address.
Proxy Configuration	Allows selection of either *No Proxy* or *System Proxy*. The System Proxy is defined by the operating system. The No Proxy setting is the default and should suffice for most installations, especially for only local network access.
Data Log File	Use the "Choose File..." button to specify a new, or select an existing, data log file. New data will be *appended* to the file at the specified location. The format of the log file is a *comma-separated values* format (.CSV) that is compatible with Microsoft Excel as well as many other spreadsheet or plotting applications. If the field is empty, logging is disabled.

3.2 Plot Settings



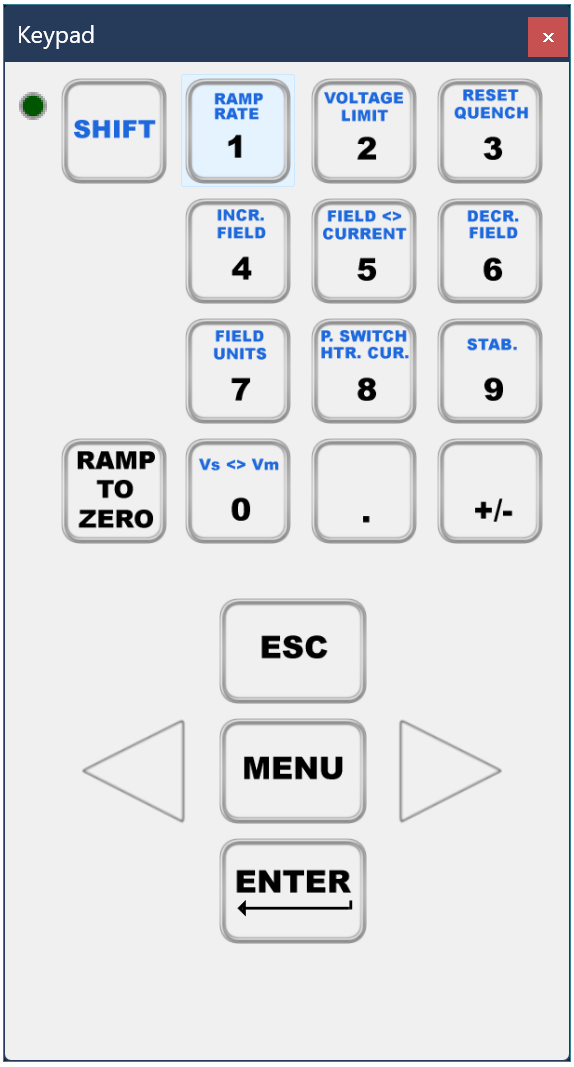
The Plot Settings configuration panel is accessible by using the Toolbar icon. The panel can be docked to either the left or right side of the Main Window, or it can be dragged off the Main Window to "float" on the desktop. The fields and interface items are described below:

X Max	The maximum time value displayed on the horizontal axis of the plot in the selected X-Axis Time Base units.
X Min	The minimum time value displayed on the horizontal axis of the plot in the selected X-Axis Time Base units. The time value starts at zero when the instrument initially enters the "Connected" state.
Y Max	The maximum field and/or current value displayed in the left vertical axis of the plot.
Y Min	The minimum field and/or current value displayed in the left vertical axis of the plot.
V Max	The maximum voltage value displayed in the right vertical axis of the plot.
V Min	The minimum voltage value displayed in the right vertical axis of the plot.
Reset Graph	Pressing this button will reset the plot area to the specified "Default Axes Limits" as described above. However, the horizontal time axis value will continue to increment.

	The time value is only reset to zero when the instrument initially enters the "Connected" state.
X-Axis Time Base	Selected either seconds or minutes for the horizontal plot axis units
Autoscroll Range	If this is selected, then the horizontal axis of the plot is automatically scrolled in time increments set by the value of (X Max - X Min). If you wish to scroll back in time to observe past data, temporarily turn off (i.e. uncheck) Autoscroll Range. A quick-access icon is provided at the bottom left of the plot area as a shortcut to toggling this setting ON/OFF.
Magnet Field (kG or T)	If checked, the magnet field will be plotted in the selected instrument field units.
Magnet Current (A)	If checked, the magnet current will be plotted in amperes.
Show Ramp Reference (A)	If checked, the noiseless, internal digital ramp reference in the Model 430 is plotted in amperes. The value is the current reference to which the Model 430 attempts to control the magnet current.
Magnet Voltage (V)	If checked, the magnet voltage will be plotted in volts. The right vertical axis sets the voltage scale in the plot. This voltage is typically measured by voltage taps across the magnet coil.

Supply Current (A)	If checked, the power supply current is plotted in amperes. Note that the power supply current and magnet current can differ if the magnet is in "persistent mode".
Supply Voltage (V)	If checked, the power supply output voltage is plotted in volts. The right vertical axis sets the voltage scale in the plot. This voltage may differ from the magnet voltage by the additional voltage drops from lead resistance and/or energy absorbers.

3.3 Keypad



The Keypad panel is accessible using the Toolbar icon. The panel can be docked on the left or right of the Main Window, or it can be dragged out of the Main Window to "float" on the desktop. The interface items are identical in function to the physical keypad of the instrument with only a couple of exceptions as described below:

Emulates the SHIFT key function. When pressed, the LED energizes to the left of the button indicating an active SHIFT state, unless the SHIFT state is already active in which case it is canceled and the LED de-energizes.

Emulates the "1" key function. When SHIFT is active, pressing the key displays the RAMP RATE settings.

Emulates the "2" key function. When SHIFT is active, pressing the key displays the VOLTAGE LIMIT setting.

Emulates the "3" key function. When SHIFT is active, pressing the key clears any active QUENCH state.

Emulates the "4" key function. When SHIFT is active, pressing the key causes the magnet field or current to increase in value.

Emulates the "5" key function. When SHIFT is active, pressing the key toggles the display between current and field units.

Emulates the "6" key function. When SHIFT is active, pressing the key causes the magnet field or current to decrease in value.

FIELD UNITS 7	Emulates the "7" key function. When SHIFT is active, pressing the key displays the **Setup > Misc > Field Units** selection menu.
P. SWITCH HTR. CUR. 8	Emulates the "8" key function. When SHIFT is active, pressing the key displays the **Setup > PSwitch > PSwitch Current (mA)** selection menu.
STAB. 9	Emulates the "9" key function. When SHIFT is active, pressing the key displays the **Setup > Load > Stability Mode** selection menu. If the Stability Mode is "Manual", then the arrow keys can be used to move to the **Setup > Load > Stability Setting (%)** entry menu.
Vs <> Vm 0	Emulates the "0" key function. When SHIFT is active, pressing the key toggles the display between supply voltage (Vs) and magnet voltage (Vm) values.
.	Emulates the decimal key function. *No SHIFT function is emulated for this key press.*
+/-	Emulates the sign change key function. *No SHIFT function is emulated for this key press.*
RAMP TO ZERO	Emulates the "Ramp to Zero" key function.
ESC	Emulates the ESC (i.e. escape) key function. The SHIFT state function is available as described in the Model 430 manual.
MENU	Emulates the MENU key function. The SHIFT state function is available as described in the Model 430 manual.

ENTER	Emulates the ENTER key function.
◁	Emulates the left-arrow key function.
▷	Emulates the right-arrow key function.

Advanced Features

4 Advanced Features

4.1 Command Line Options

The application supports several command line options. These can be specified, for example, on the Windows platform by creating a custom shortcut and adding the options in the Target field of the shortcut Properties.

Command line option	Meaning
-h	Hide (minimize) the Magnet-DAQ interface on launch.
-x	Label the Magnet-DAQ window as "X Axis" and superimpose an X on the app icon.
-y	Label the Magnet-DAQ window as "Y Axis" and superimpose a Y on the app icon.
-z	Label the Magnet-DAQ window as "Z Axis" and superimpose a Z on the app icon.
-p	Start the stdin/stdout parser function (for Python or QProcess use). *Firmware upgrades are not allowed in this mode.* If this option is specified, the data rate of the connected Model 430 is fixed at one sample per second.
-a *address*	Start and auto-connect to IP *address*.
--port *portid*	Connect to specified *port number* (for proprietary simulation use only).
--telnet *portid*	Show display echoed to specified *port number* (for proprietary simulation use only).

Example:

"C:\Program Files\American Magnetics, Inc\Magnet-DAQ\Magnet-DAQ.exe" -h -z -p -a 192.168.1.14

This starts the application in minimized mode, labels the Main Window and taskbar icon with the letter Z (for Z-axis), starts the stdin/stout parser, and autoconnects to the device at IP address 192.168.1.14.

NOTE: The --port and --telnet options are designed to support AMI-proprietary superconducting magnet and Model 430 simulation programs that can co-exist on the same host for testing, debugging, and demo purposes. They should not be used by customers in typical applications.

4.2 Python/Remote Scripting Support

The Magnet-DAQ application supports a parsing mode (specified by using the -p command line option on launch) that allows communication with the application using the *stdin* and *stdout* interfaces. This can be utilized by Python-based scripts or the QProcess C++ object that is part of the Qt Toolkit. These are only examples — any method that can utilize *stdin* and *stdout* can be used to communicate.

NOTE: The *Multi-Axis Operation* application distributed by AMI for operating multi-axis superconducting magnet systems uses the parser function of Magnet-DAQ and QProcess objects to operate up to three separate Model 430 units in tandem.

This interface allows reading and writing a subset of the Model 430 settings and state *without* having to deal with TCP/IP sockets or serial port programming. Not all features of the Model 430 are, as of yet, exposed to this parsing interface. The specific commands and queries that are presently supported are shown below.

The lower case characters in commands and queries indicate the *optional* long form of the command/query. The uppercase letters are the short form. All decimal values are returned with 10 significant digits to the right of the decimal. All return values are terminated with a linefeed character, '\n'.

Command parameters shown in braces { } indicate valid arguments (choose *one*). Parameters shown in < > are required decimal arguments, where ( ) indicates value units if applicable. When sending commands, omit all { } < > ( ) characters. Commands do not return a value.

NOTE: Parser errors will be enunciated with a beep by the application. An error queue is also generated as a last-in-first-out (LIFO) list. Use the **`SYSTem:ERRor?`** and **`SYSTem:ERRor:COUNt?`** queries to check the error status. It is recommended to add the error checking as a standard process of sending a command to the parser.

Commands

SYSTEM COMMANDS

`*CLS`

Clears the parser error queue.

`EXIT`

Disconnects from any connected Model 430 and gracefully exits the application.

CONTROL COMMANDS

`RAMP`

Places the Model 430 Programmer in automatic ramping mode. The Model 430 will continue to ramp at the configured ramp rate(s) until the target field/current is achieved.

`PAUSE`

Pauses the Model 430 Programmer at the present operating field/current.

`PSwitch {0|1}`

Turns the persistent switch heater OFF and ON. Sending “0” turns the switch heater OFF. Sending a “1” turns the switch heater ON. The default value is “0”.

`ZERO`

Activates ramping to zero current/field without altering the Target Setpoint value.

CONFIGURATION COMMANDS

`CONFigure:CURRent:LIMit <current (A)>`

Sets the Current Limit in amperes. The Current Limit is the largest magnitude operating current allowed during any ramping mode. For bipolar power supplies, the Current Limit functions as both a positive and negative current limit. This value is typically provided by the magnet manufacturer as the maximum safe limit of magnet operation.

`CONFigure:CURRent:TARGet <current (A)>`

Sets the target setpoint current in amperes.

`CONFigure:COILconst <value (kG/A, T/A)>`

Sets the coil constant (also referred to as the field-to-current ratio) per the selected field units. The coil constant must be set to a non-zero, positive value in order to command or query the MagnetDAQ app in units of field.

CONFigure:FIELD:TARGet <*field (kG, T)*>

Sets the target field in units of kilogauss or tesla, per the selected field units. This command requires that a non-zero coil constant be defined, otherwise the command is ignored and a device error will be generated.

CONFigure:FIELD:UNITS {0|1}

Sets the preferred field units. Sending "0" selects kilogauss. A "1" selects tesla. "0" is the default value. The selected field units are applied to both the MagnetDAQ display and the related parser commands.

CONFigure:INDuctance <*inductance (H)*>

Sets the currently-connected magnet inductance in Henries.

CONFigure:PSwitch {0|1}

"0" indicates that a persistent switch is not installed on the connected superconducting magnet. "1" indicates that a persistent switch is installed. The default value is "1". If a persistent switch is installed, the persistent switch heated current should be specified. Heating/cooling times should also be specified for timer-based switch transitions.

CONFigure:PSwitch:CURRent <*current (mA)*>

Sets the persistent switch heater current in mA.

CONFigure:PSwitch:TRANsition {0|1}

Sending a value of "0" selects the timer-based switch transition detection. Sending "1" selects the magnet voltage-based transition detection. Timer-based transition detection is the default.

CONFigure:PSwitch:HeatTIME <*time (sec)*>

Sets the time required in seconds for the persistent switch to become resistive after the persistent switch heater has been activated.

CONFigure:PSwitch:CoolTIME <*time (sec)*>

Sets the time required in seconds for the persistent switch to become superconducting after the persistent switch heater has been deactivated.

CONFigure:PSwitch:PowerSupplyRampRate <*rate (A/s)*>

Sets the ramp rate that will be used by the power supply to ramp the current during the PERSIST. SWITCH CONTROL operation when the switch is in a cooled state. This ramp rate can

be much higher than when the switch is heated and the magnet is in the circuit. For more information as to how this function operates, refer to the Model 430 manual.

CONFigure:PSwitch:CoolingGAIN <*percent*>

Sets the persistent switch cooling gain in percent.

CONFigure:RAMP:RATE:FIELD <*segment*>,<*rate (kG/s, kG/min, T/s, T/min)*>,<*upper bound (kG, T)*>

Sets the ramp rate for the specified segment (values of 1 through the defined number of ramp segments are valid) in units of kilogauss/second or minute, or tesla/second or minute (per the selected field units and ramp rate units), and defines the field upper bound for that segment in kilogauss or tesla (see the Model 430 manual for details of the use of ramp segments). This command requires that a non-zero coil constant be defined; otherwise, the command is ignored and a device error will be generated.

CONFigure:RAMP:RATE:CURRent <*segment*>,<*rate (A/s, A/min)*>,<*upper bound (A)*>

Sets the ramp rate for the specified segment (values of 1 through the defined number of ramp segments are valid) in units of A/sec or A/min (per the selected ramp rate units), and defines the current upper bound for that segment in amperes (see the Model 430 manual for details of the use of ramp segments).

CONFigure:RAMP:RATE:UNITS {0|1}

Sets the preferred ramp rate time units. Sending “0” selects seconds. A “1” selects minutes. “0” is the default value. The selected units are applied to both the MagnetDAQ display and the related parser commands.

CONFigure:RAMP:RATE:SEGments <*value*>

Sets the number of ramp segments from 1 to 10. 1 is the default. See the Model 430 manual for details of the use of ramp segments.

CONFigure:STABility:MODE {0|1|2}

“0” configures the stability mode as “Auto”, “1” is “Manual”, and “2” is “Test”. If “Auto” mode is selected and an inductance value is not specified (i.e. the inductance entry is zero), the MagnetDAQ app will beep once, generate an error, and revert to “Manual” stability mode. Manual mode is the default value. In “Auto” stability mode, the stability setting is automatically adjusted per the presence or absence of a persistent switch, the presence or absence of a stabilizing resistor, and the estimated or measured magnet inductance. See the Model 430 manual for details.

CONFigure:STABility <*percent*>

Sets the stability setting in percent. Valid range is 0.0 to 100.0%.

```
CONFigure:STABility:RESistor {0|1}
```

An argument of “0” specifies no stabilizing resistor is installed. “1” indicates a stabilizing resistor is installed. “0” is the default value.

```
CONFigure:VOLTage:LIMit <voltage (V)>
```

Sets the ramping Voltage Limit in volts. The limit may not exceed the maximum output voltage of the power supply.

Queries

SYSTEM QUERIES

```
*IDN?
```

Returns the identification string of the Magnet-DAQ application. The identification string contains the app name and revision level of app. Example output: **`MagnetDAQ,1.05`**

```
SYSTem:ERRor?
```

Returns the most recent system error added to the error queue and removes it from the queue. If the queue is empty, the return string is `0,"No errors"`. See the Error Codes section below.

```
SYSTem:ERRor:COUNt?
```

Returns the integer number of errors currently in the system error queue.

STATE/CONFIGURATION QUERIES

```
STATE?
```

Returns an integer value corresponding to the ramping state according to the table below:

Return Values for `STATE?` Query	
Return Value	Meaning
1	RAMPING to target field/current
2	HOLDING at the target field/current
3	PAUSED
4	Ramping in MANUAL UP mode

5	Ramping in MANUAL DOWN mode
6	ZEROING CURRENT (in progress)
7	Quench detected
8	At ZERO current
9	Heating persistent switch
10	Cooling persistent switch
11	External Rampdown active

`CURRent:LIMit?`

Returns the Current Limit in amperes.

`CURRent:MAGnet?`

Returns the current flowing in the magnet in amperes. If the magnet is in persistent mode, the command returns the current that was flowing in the magnet when persistent mode was last entered.

`CURRent:SUPPly?`

Returns the measured power supply current in amperes.

`CURRent:TARGet?`

Returns the target current setting in amperes.

`CURRent:REFerence?`

Returns the noiseless, internal Model 430 ramp reference value in amperes.

`COILconst?`

Returns the coil constant setting in kG/A or T/A per the selected field units.

`FIELD:MAGnet?`

Returns the calculated field in kilogauss or tesla, per the selected field units. This query requires that a non-zero coil constant be defined; otherwise, the query is ignored and an error is generated in the error queue. The field is calculated by multiplying the measured magnet current by the coil constant. If the magnet is in persistent mode, the command returns the field that was present when persistent mode was last entered.

`FIELD:TARGet?`

Returns the target current setting in kilogauss or tesla, per the selected field units. This query requires that a non-zero coil constant be defined; otherwise, the query is ignored and an error is generated in the error queue.

FIELD:UNITS?

Indicates the field units. Returns a value of "0" for kilogauss units or "1" for tesla units.

PERSistent?

Indicates the persistence state of a connected magnet. Returns a value of "0" for not persistent, or "1" for persistent. The switch heater must be in the OFF state for a magnet to become persistent. There may be some delay between the OFF switch heater state and persistence indication, per the switch cooling time parameter.

PSwitch?

Indicates the state of the persistent switch heater. Returns a value of "0" for OFF or "1" for ON.

PSwitch:CURRent?

Returns the persistent switch heater current in mA.

PSwitch:TRANsition?

A return value of “0” indicates the timer-based switch transition detection. A return value of “1” indicates the magnet voltage-based transition detection.

PSwitch:HeatTIME?

Returns the time required in seconds for the persistent switch to become resistive after the persistent switch heater has been activated.

PSwitch:CoolTIME?

Returns the time required in seconds for the persistent switch to become superconducting after the persistent switch heater has been deactivated.

PSwitch:PowerSupplyRampRate?>

Returns the ramp rate that will be used by the power supply to ramp the current during the PERSIST. SWITCH CONTROL operation when the switch is in a cooled state. This ramp rate can be much higher than when the switch is heated and the magnet is in the circuit. For more information as to how this function operates, refer to the Model 430 manual.

PSwitch:CoolingGAIN?

Returns the persistent switch cooling gain in percent.

QUench:CURRent?

Returns the current value captured at the time of most recent quench detection in amperes.

RAMP:RATE:FIELD:<*segment*>?

Returns the ramp rate for the specified segment (values of 1 through the defined number of ramp segments are valid) in units of kilogauss/second or minute, or tesla/second or minute (per the selected field units and ramp rate units), and returns the field upper bound for that segment in kilogauss or tesla (see the Model 430 manual for details of the use of ramp segments). This command requires that a coil constant be defined; otherwise the query is ignored and an error is generated in the error queue.

Example (returns 0.015 kG/s with an upper bound of 10.0 kG for ramp segment #2):

```
RAMP:RATE:FIELD:2?
0.015,10.0
```

RAMP:RATE:CURRent:<*segment*>?

Returns the ramp rate for the specified segment (values of 1 through the defined number of ramp segments are valid) in units of A/sec or A/min (per the selected ramp rate units), and returns the current upper bound for that segment in amperes (see the Model 430 manual for details of the use of ramp segments).

Example (returns 0.136612 A/s with an upper bound of 47.8 A for ramp segment #1):

```
RAMP:RATE:CURR:1?
0.136612,47.8
```

RAMP:RATE:UNITS?

Returns the chosen ramp rate timebase units. A “0” indicates seconds. A “1” indicates minutes.

RAMP:RATE:SEGments?

Returns the number of ramp segments from 1 to 10. See the Model 430 manual for details of the use of ramp segments.

STABility:MODE?

“0” indicates the present stability mode as “Auto”, “1” is “Manual”, and “2” is “Test”. In “Auto” stability mode (the preferred setting), the stability setting is automatically adjusted per the presence or absence of a persistent switch, the presence or absence of a stabilizing resistor, and the estimated or measured magnet inductance. See the Model 430 manual for details.

STABility?

Returns the stability setting in percent. Valid range is 0.0 to 100.0%.

STABility:RESistor?

An return value of “0” specifies no stabilizing resistor is installed. “1” indicates a stabilizing resistor is installed.

`VOLTage:LIMit?`

Returns the Voltage Limit in volts. The Voltage Limit is applied to the power supply output as not every magnet manufacturer/integrator provides magnet voltage taps.

`VOLTage:MAGnet?`

Returns the voltage (V) across the magnet if voltage taps are available and connected to the Model 430.

`VOLTage:SUPPly?`

Returns the presently commanded power supply voltage in volts.

Error Codes

The following are error codes returned by the **`SYSTem:ERRor?`** query. The queue is last-in, first-out (LIFO). In addition to the errors defined for the parser function of the application as shown below, *device errors* generated by the resulting communication with the selected Model 430 device are also accumulated in this error queue. The device error codes are described in detail in the Model 430 manual in the *Remote Interface Reference : Error Messages* section.

At present is it not possible to limit the error queue to those generated directly by the parsing interface and the resulting device communication, therefore the error queue contains *all device errors* generated while connected.

`0,"No error"`

The error queue is empty.

COMMAND ERRORS

`-101,"Unrecognized command"`

The command is unrecognized.

`-102,"Invalid argument"`

An argument required for the command is invalid.

`-103,"Non-boolean argument"`

A non-boolean value is provided to a command requiring a boolean argument.

```
-104,"Missing parameter"
```

A required parameter is missing from the command argument list.

```
-105,"Value out of range"
```

A parameter is out of range.

```
-106,"Undefined coil const"
```

Attempted to command or query in field units with a zero (undefined) coil constant value. Specify a coil constant.

```
-151,"Non-numerical entry"
```

A parameter is in a non-numerical format.

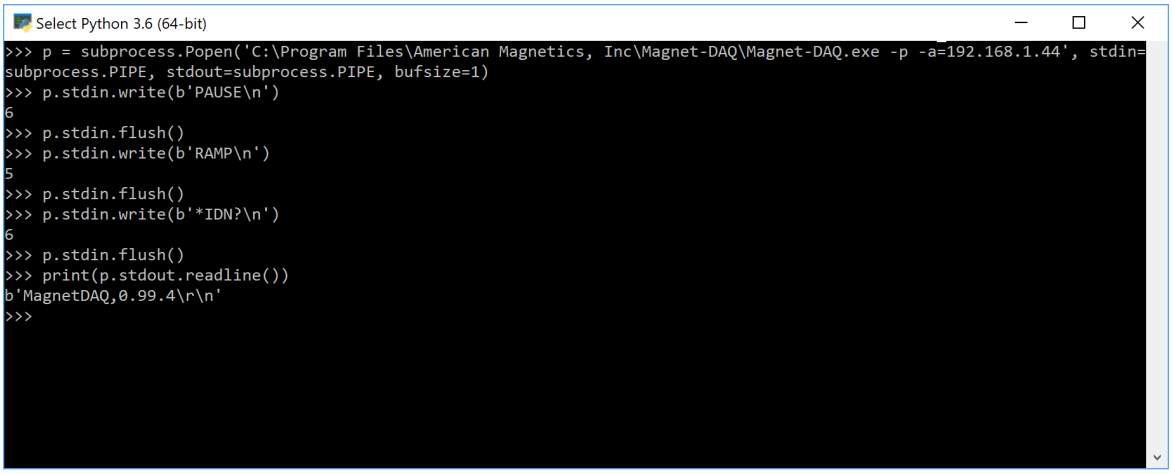
QUERY ERRORS

```
-201,"Unrecognized query"
```

The query is unrecognized.

Python Example

The image below illustrates an interactive Python session from Visual Studio 2017 that launches the Magnet-DAQ application with the parser function enabled, autoconnects to the device at IP address 192.168.1.44, sends the **PAUSE** command, sends the **RAMP** command, and then sends the ***IDN?** query and reads the result.



4.3 Table Import Format

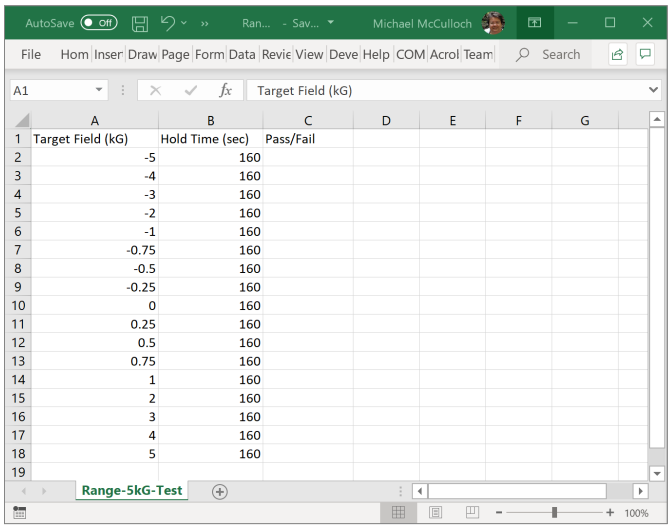
The Table tab format is Comma-Separated Values (CSV) that is compatible with Excel (or other spreadsheet applications) as well as a simple text editor such as Notepad++.

The format is a file with a single-line, four column header and then subsequent rows of target values in current (A) or field (kG or T) units, and optional Hold Time for each in seconds. Omission of the Hold Time value results in a zero Hold Time after the import.

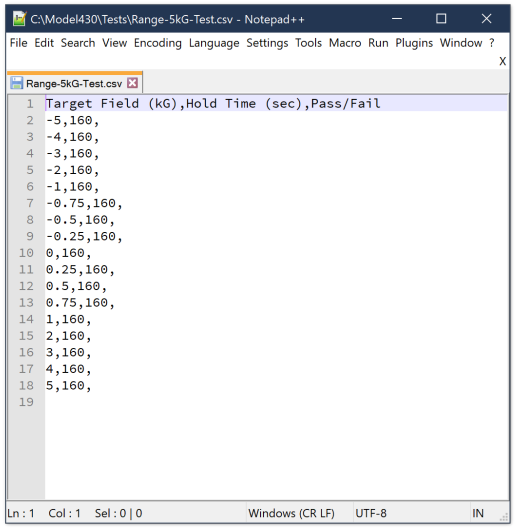
The format is illustrated below with a spreadsheet view on the left (in Microsoft Excel) and a simple text editor view of the *same file* on the right. As can be observed in the illustration, the CSV format is a simple text, comma-separated format.

Example below: The header text should be as shown below with the exception that if you wish to use tesla units for the field magnitude, change (kG) to (T). Specifying the units allows the vector table load function to properly identify (and convert if necessary) the field values.

NOTE: By default, omitting any units (A, kG, or T) in the header assumes Amperes for all target values.



Target Field (kG)	Hold Time (sec)	Pass/Fail
-5	160	
-4	160	
-3	160	
-2	160	
-1	160	
-0.75	160	
-0.5	160	
-0.25	160	
0	160	
0.25	160	
0.5	160	
0.75	160	
1	160	
2	160	
3	160	
4	160	
5	160	



```
Target Field (kG),Hold Time (sec),Pass/Fail
-5,160,
-4,160,
-3,160,
-2,160,
-1,160,
-0.75,160,
-0.5,160,
-0.25,160,
0,160,
0.25,160,
0.5,160,
0.75,160,
1,160,
2,160,
3,160,
4,160,
5,160,
```

Python Examples

5 Python Examples

5.1 Example Python Script

The following is the Example.py Python script as referenced in the Table Tab execution example. This example supports any digital multimeter that is HP/Agilent 34401A compatible, which includes most recent Keysight models.

This simple example script accepts one argument for the IP address of the digital multimeter. It then specifies a file path for saving data, connects to the instrument, and commands the instrument to average 60 voltage samples and return the average value. The return value is then appended to the file and the script exits.

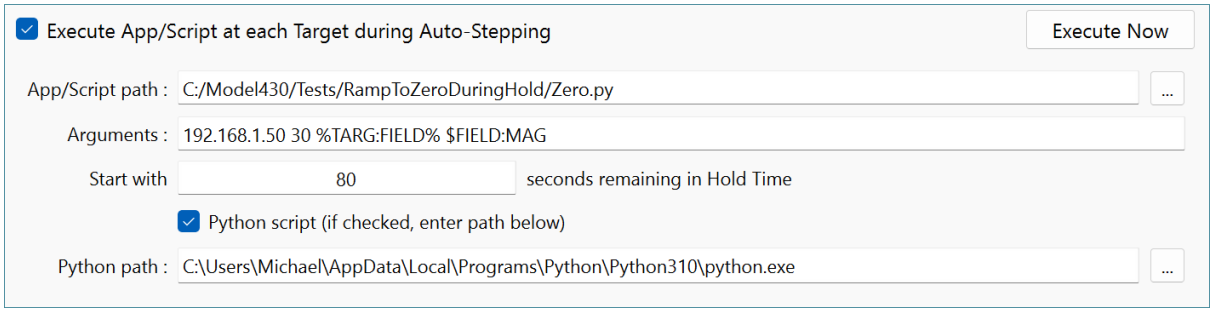
```
# This is an example Python 3.x script showing how to connect
# to a Keysight 34401A compatible DMM and retrieve an average of
# the voltage. The value is written to a file (see "path" below).
# Change the path to a location where your account has file write
# privileges.

import sys
import socket

# open socket to Keysight 34401A compatible DMM
DVM = socket.socket()

# check to see if arguments provided, should be one arg with ip address
if len(sys.argv) > 1:
    DVM.connect((str(sys.argv[1]), 5025))
else:
    DVM.connect(("192.168.1.17", 5025))    # default if not provided

#echo instrument id
DVM.send(b'*IDN?\n')
data = DVM.recv(1024)
print(data.decode('utf-8'))

# append data to a file
path = 'C:/Model430/Tests/Example.csv'
datafile = open(path, 'a')

print((b'Getting 10 second average of voltage...').decode('utf-8'))

# get 60 sample averages (about 10 sec)
DVM.send(b'CONF:VOLT:DC AUTO\n')
DVM.send(b'VOLT:DC:NPLC 10\n')
DVM.send(b'TRIG:SOUR BUS\n')
DVM.send(b'SAMP:COUN 60\n')
DVM.send(b'CALC:AVER:STAT ON\n')
DVM.send(b'INIT\n')
DVM.send(b'*TRG\n')
DVM.send(b'*WAI\n')
```

```
DVM.send(b'CALC:AVER:AVER?\n')
data = DVM.recv(1024)

# write to file
datafile.write(data.decode("utf-8").strip('\n'))
datafile.write('\n');

# close datafile
datafile.close()

# close socket
DVM.close()
```

5.2 Example Python Script with Special Variables

The following is the RampToZero.py Python script using special variables.

This simple example script ramps the power supply system to zero current after the persistent switch is cooled, holds for a specified period (in arg[2]), and then ramps the current back to the target value before the switch is reheated. The special arg values (%TARG:FIELD% and $FIELD:MAG) are output and the script exits. An example Magnet-DAQ Table tab configuration that matches this example is:




```
import socket
import time
import sys

# open socket to Model 430
Model430 = socket.socket()
Model430.connect((sys.argv[1], 7185))

# first check that switch heater is OFF
Model430.send(b'PS?\n')
data = Model430.recv(1024)

if data != b'0\r\n':
```

```
   print("Script aborted: Not in persistent mode")
   sys.exit(0)

# proceed with ramping power supply to zero
Model430.send(b'ZERO\n')

data = b'10\n'

# wait until ramp to zero completes
while data != b'8\r\n':
   time.sleep(1)
   Model430.send(b'STATE?\n')
   data = Model430.recv(1024)

   # check if switch cooling step failed
   if data != b'6\r\n' and data != b'8\r\n':
      print("Script aborted: Switch cooling failed")
      sys.exit(1)

   #check for magnet voltage indicating switch cooling failed
   Model430.send(b'VOLT:MAG?\n')
   Vm = float(Model430.recv(1024))

   if abs(Vm) > 1.0:
      Model430.send(b'PAUSE\n')
      time.sleep(2)
      # turn on switch heater
      Model430.send(b'PS 1\n')
      print("Script aborted: Switch cooling failed")
      sys.exit(1)

# enter paused state
Model430.send(b'PAUSE\n')

# wait argv[2] seconds and RAMP
time.sleep(int(sys.argv[2]))

Model430.send(b'RAMP\n')

# close sockets
Model430.close()

# successful
# print the target field and magnet field passed as args (units of Tesla
print(sys.argv[3] + "T," + sys.argv[4] + "T")

sys.exit(0)
```

Support

6 Support

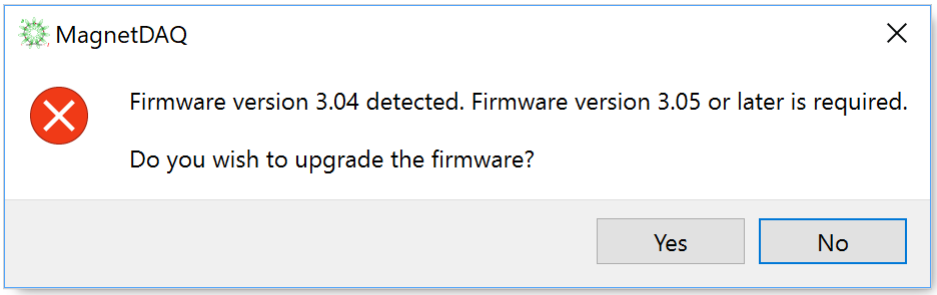
6.1 Firmware Upgrade Wizard

The Magnet-DAQ application contains firmware udpates for the Model 430 Programmer. Each release of Magnet-DAQ may add new features that do not work without a firmware upgrade. Furthermore, firmware upgrades may be recommended due to corrected bugs.

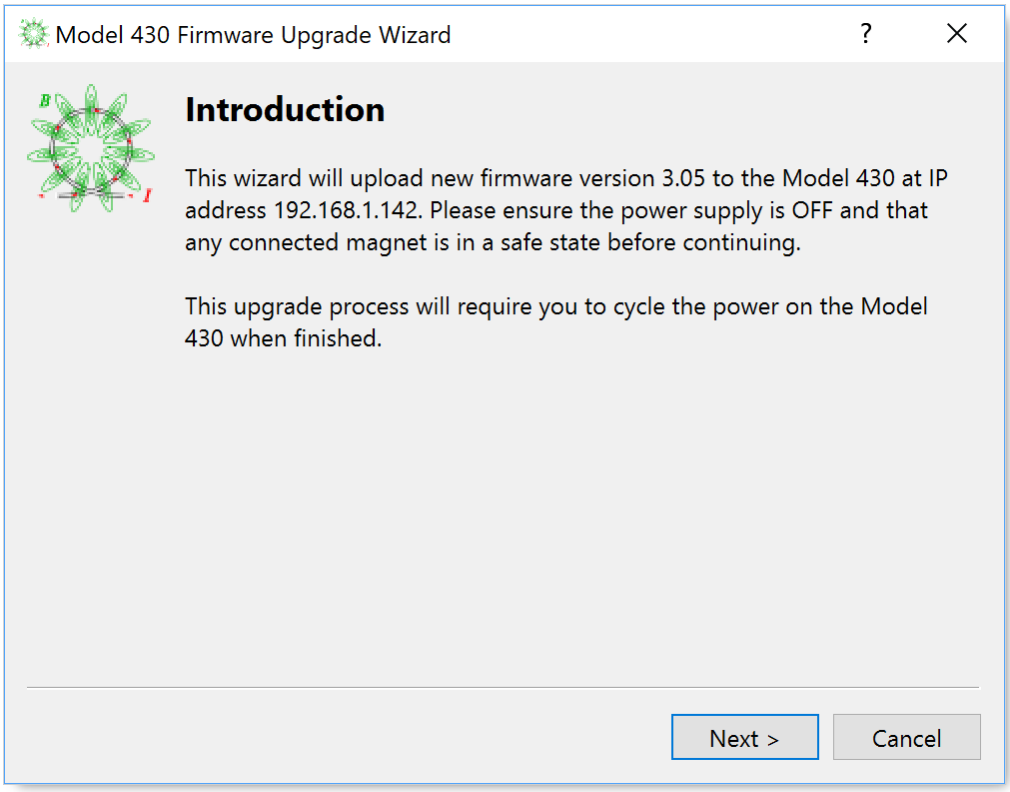
Therefore, Magnet-DAQ may *require* a firmware upgrade of the Model 430 in order to function. The firmware upgrade is entirely *optional*, but Magnet-DAQ will refuse to operate with the target device without the upgrade.

NOTE: If you have any concerns about a firmware upgrade, please feel free to contact AMI Technical Support before attempting the upgrade. At present, the Firmware Upgrade Wizard only supports upgrades for Model 430 Programmers running version 1.62 or later firmware. Earlier versions may be supported in future Magnet-DAQ releases with further regression testing of legacy devices by AMI to ensure compatibility. A manual upgrade process can be detailed for pre-1.62 firmware systems by contacting AMI Technical Support.

If a firmware upgrade is required, Magnet-DAQ will first present the user a notice (the version numbers may be different with your system and each firmware release):



Upon the selection of "Yes", the Firmware Upgrade Wizard will appear as illustrated below (again the version numbers and IP address may change):



NOTE: Please carefully observe the wizard instructions during the upgrade process. Cycling power on the Model 430 will be required in order to complete the upgrade process.

The upgrade process transfers the firmware by standard FTP protocol (ports 20 and 21). If your local network blocks internal FTP traffic, the update will not be accomplished and the wizard will display an error.

6.2 Updates

The Magnet-DAQ is provided as a open-source application subject to the GPL verison 3 or later license.

The latest source code and binaries distributions can be found on the BitBucket Cloud at:

https://bitbucket.org/americanmagneticsinc/magnet-daq

Ready-to-use binaries are provided on the Downloads page for 64-bit versions of Windows 7 or later, Linux (Ubuntu 14.04 or later recommended), and Apple macOS Sierra or later.

Users are encourage to check the site often for updates (check out the "Watchers" feature). Notification lists will not be maintained by AMI.

License Terms

7 License Terms

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of

such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and

control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest

had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's “contributor version”.

A contributor's “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of

the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of

any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

- A -

- C -

- D -

- E -

- F -

- G -

- I -

- K -

- L -

- M -

- N -

- P -

- Q -

- R -

- S -

- T -

- V -

- W -

- Z -