# Multi-Axis Operation Help

# USER MANUAL


American Magnetics, Inc.

# Quick Start

# 1 Quick Start

The Multi-Axis Operation application is designed to provide a useful interface for controlling multiple AMI Model 430 Programmers for multi-axis applications. Most AMI multi-axis magnet designs incorporate orthogonal coils that correspond to 3D cartesian coordinates (X, Y, and Z). However, it can be more intuitive to work in spherical coordinates for the resulting magnetic field vector, which is supported in the application.

**The Multi-Axis Operation application requires the installation of AMI's Magnet-DAQ application as a prerequisite.** If you have not installed Magnet-DAQ, please obtain it from https://bitbucket.org/americanmagneticsinc/magnet-daq.

A typical operational flow for using the Vector Table feature is shown in the following steps. Once this is initially performed, the *File | Save Settings* menu item can be used to save the complete state of the interface to save time when reloading for subsequent sessions.

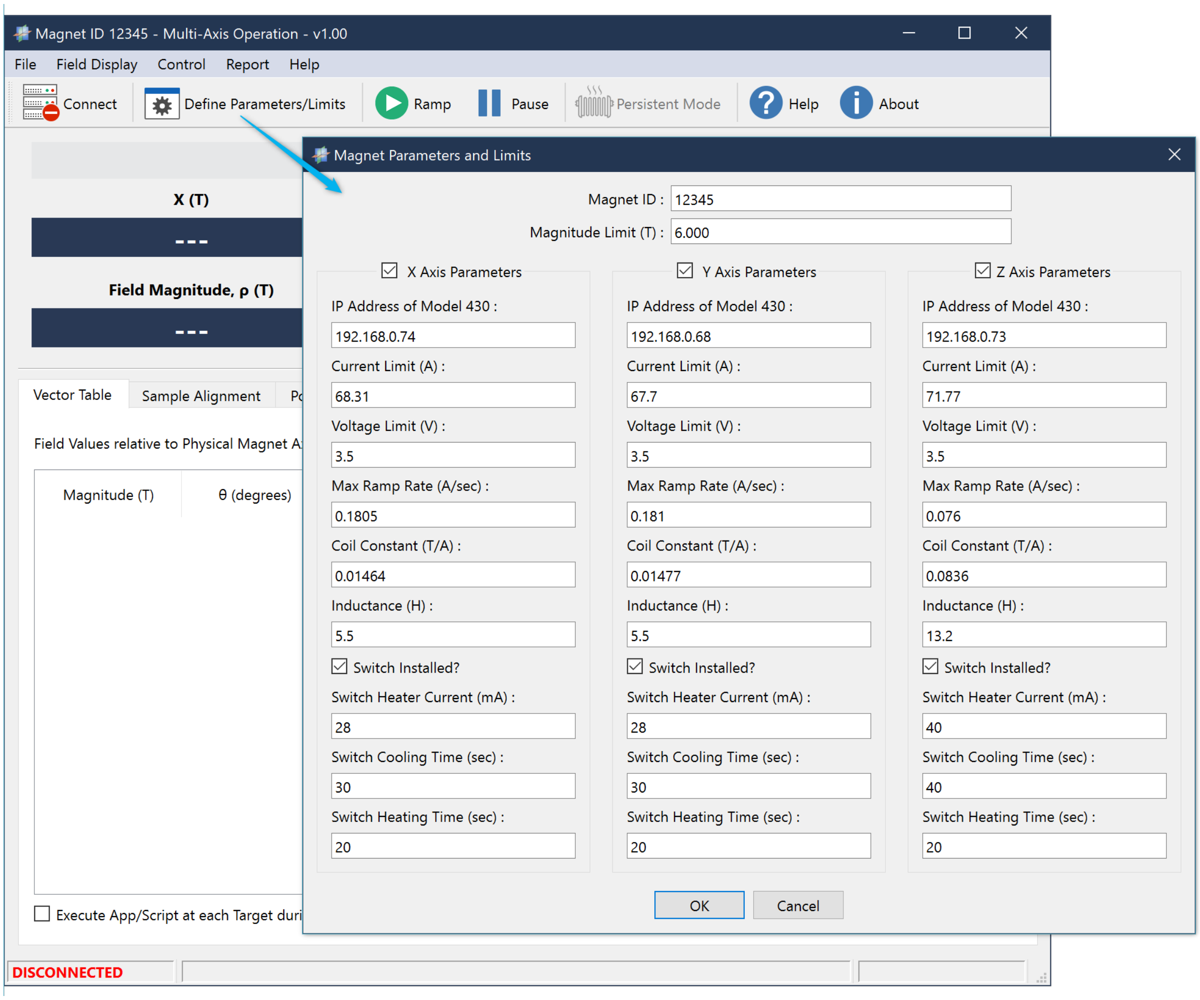
More advanced usage can be accomplished by using the Sample Alignment and Rotation in Sample Plane features. This application also provides a stdin/stout scripting feature that allows it to be externally-controlled from another application written in Python, LabVIEW, C++, etc.

## Step 1: Launch Multi-Axis Operation and open the *Define Parameters/Limits* dialog

Enter all critical parameters for the multi-axis magnet system in the *Define Parameters/Limits* dialog. These settings for each axis should be available from the magnet specification sheets, except the IP addresses which can be obtained from the *Menu* | *Net Settings* display of each Model 430.

> **NOTE:** AMI typically recommends static IP address assignment for each Model 430 to avoid DHCP moving the IP addresses between uses.

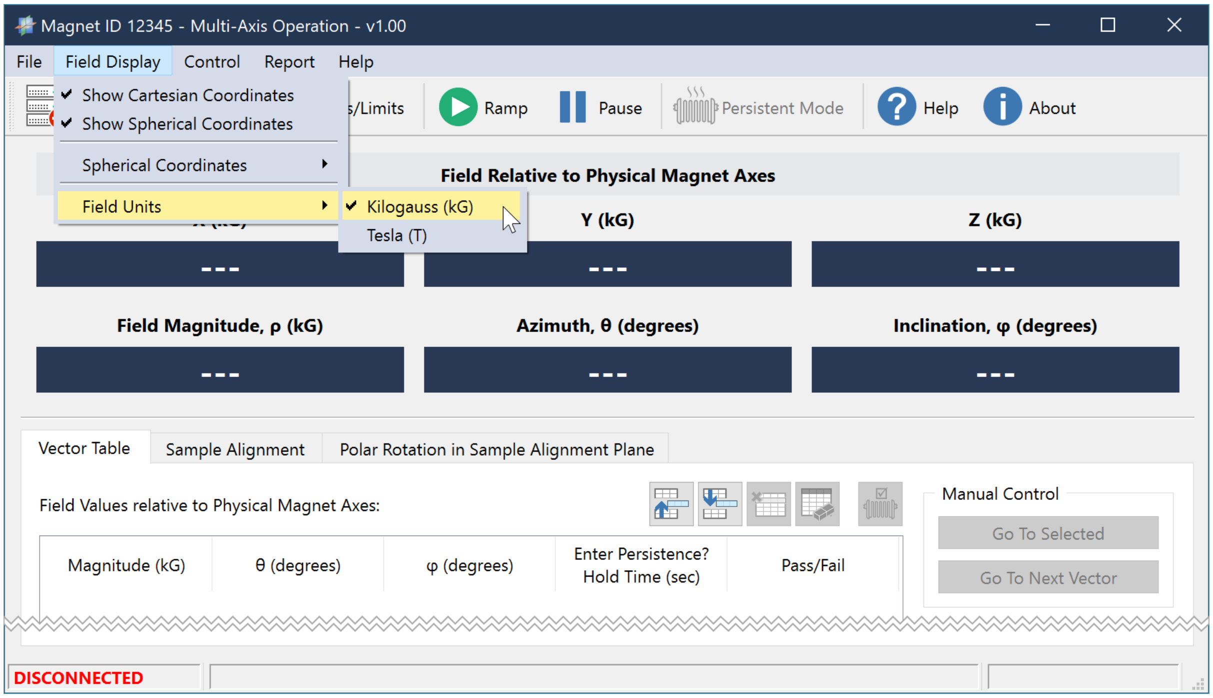
 



Magnet ID 12345 - Multi-Axis Operation - v1.00
File
Field Display
Control
Report
Help
Connect
Define Parameters/Limits
Ramp
Pause
Persistent Mode
Help
About
X (T)
---
Field Magnitude, ρ (T)
---
Vector Table
Sample Alignment
Field Values relative to Physical Magnet A
Magnitude (T)
θ (degrees)
Execute App/Script at each Target duri
DISCONNECTED
Magnet Parameters and Limits
Magnet ID : 12345
Magnitude Limit (T) : 6.000
X Axis Parameters
IP Address of Model 430 : 192.168.0.74
Current Limit (A) : 68.31
Voltage Limit (V) : 3.5
Max Ramp Rate (A/sec) : 0.1805
Coil Constant (T/A) : 0.01464
Inductance (H) : 5.5
Switch Installed?
Switch Heater Current (mA) : 28
Switch Cooling Time (sec) : 30
Switch Heating Time (sec) : 20
Y Axis Parameters
IP Address of Model 430 : 192.168.0.68
Current Limit (A) : 67.7
Voltage Limit (V) : 3.5
Max Ramp Rate (A/sec) : 0.181
Coil Constant (T/A) : 0.01477
Inductance (H) : 5.5
Switch Installed?
Switch Heater Current (mA) : 28
Switch Cooling Time (sec) : 30
Switch Heating Time (sec) : 20
Z Axis Parameters
IP Address of Model 430 : 192.168.0.73
Current Limit (A) : 71.77
Voltage Limit (V) : 3.5
Max Ramp Rate (A/sec) : 0.076
Coil Constant (T/A) : 0.0836
Inductance (H) : 13.2
Switch Installed?
Switch Heater Current (mA) : 40
Switch Cooling Time (sec) : 40
Switch Heating Time (sec) : 20
OK
Cancel

## Step 2: Choose your Field Display preferences

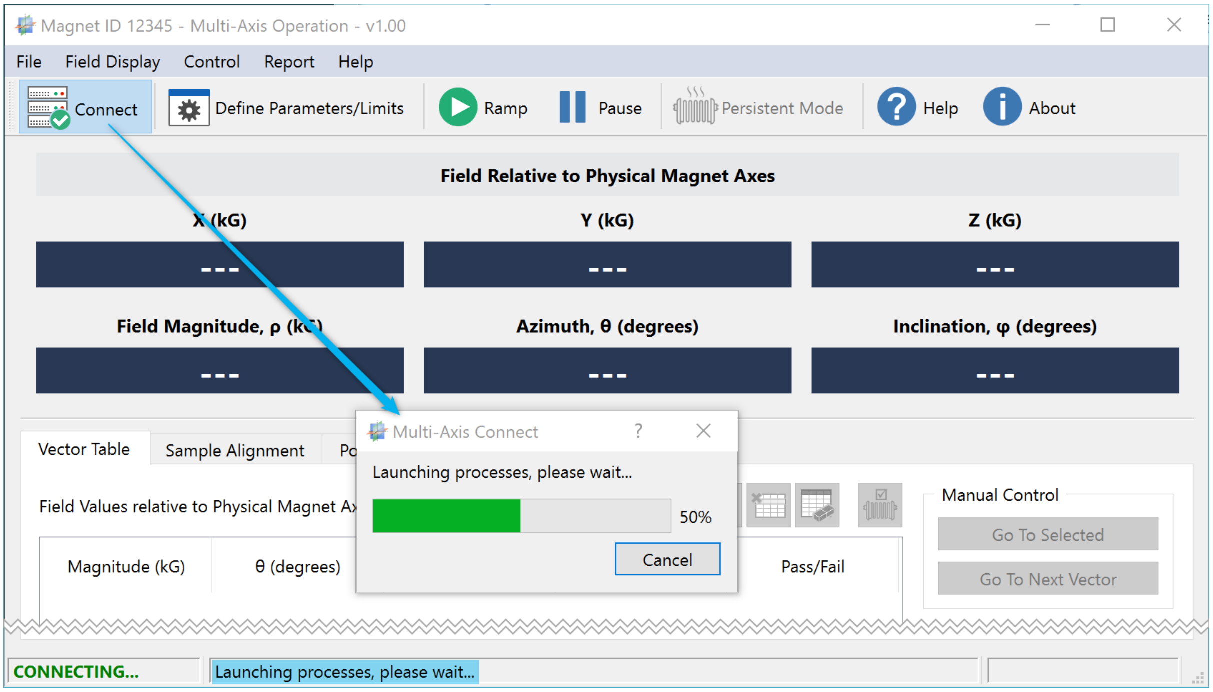
Choose your *Field Display preferences* by checking/unchecking appropriate menu bar items. AMI technicians like to work in kilogauss field units using the mathematical convention for spherical coordinates. You may prefer otherwise.



## Step 3: Use the Connect button to initiate a connection to the multi-axis system

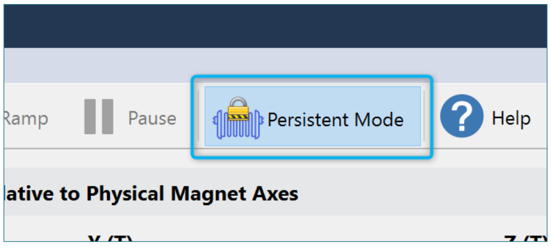
Use the *Connect* button to initiate a connection to the multi-axis system. Up to three (3) instances of Magnet-DAQ will launch in the background (the icons will be visible in the taskbar, but minimized) – one for each axis. Be patient as this takes some time. Watch the status bar at the bottom of the window for progress updates.

> **NOTE:** Do you see a firmware upgrade notice? Please see the Magnet-DAQ application Help file for more details on how to upgrade the firmware for your Model 430 Programmer. Firmware upgrades are not allowed during sessions launched by this application.




If the magnet has one or more persistent switches that are presently in a cooled state upon connection, then the operator will be first required to exit the Persistent Mode to continue. **Persistent Mode is indicated by a blue icon in the toolbar as shown below with a superimposed "lock" symbol.** Press the icon to initiate exit from the Persistent Mode.

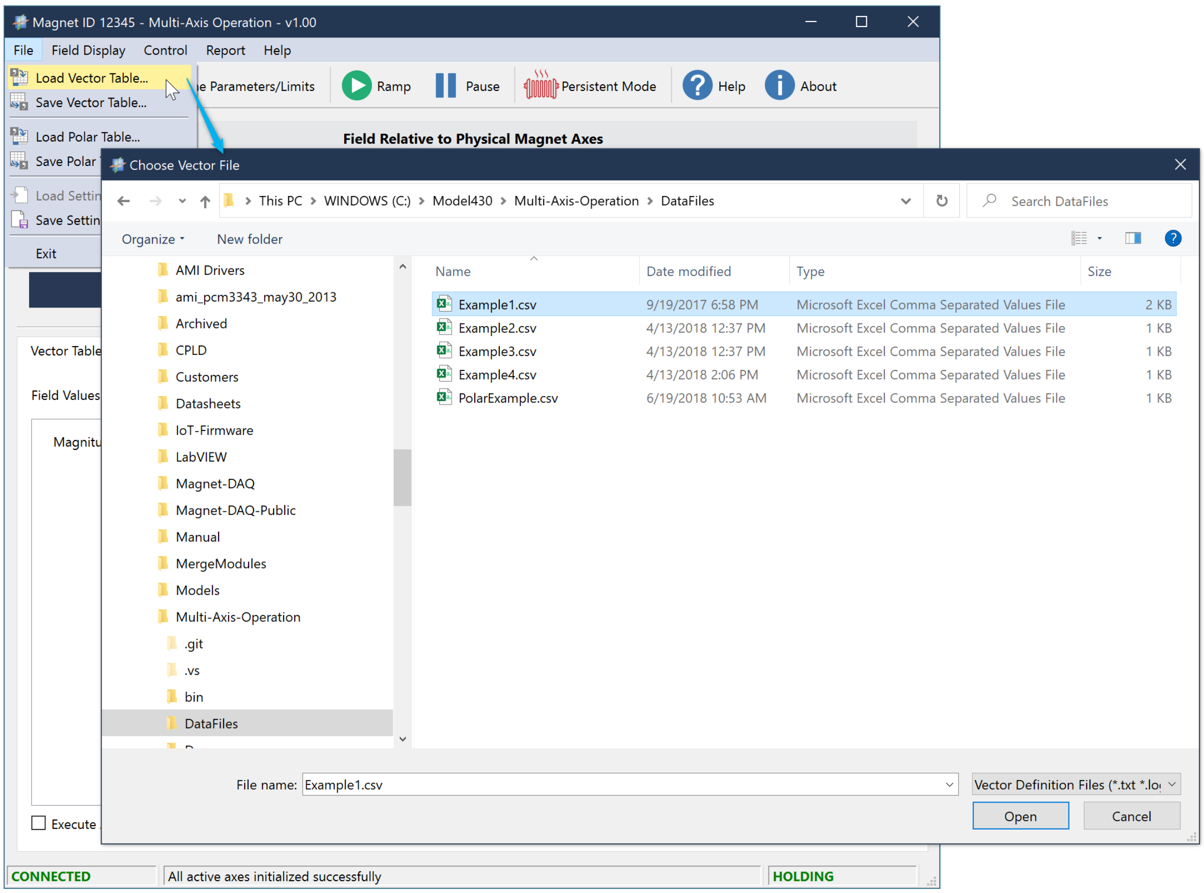
> **NOTE:** The application will first query the Model 430 units for the last known magnet current before each entered persistence and match that current *before* activating the switch heater for each axis.




*Persistent Mode indicator*

### Step 4: Use the *File | Load Vector Table...* menu item to bring up a file chooser

Create a .csv file (Excel-compatible, comma-delimited format) with a header line and then multiple rows of field values in spherical coordinates that you wish to execute. The format details and an example are provided in the Vector Table Format topic. Then, use the *File | Load Vector Table...* menu item to bring up a chooser to point to and open the .csv file.

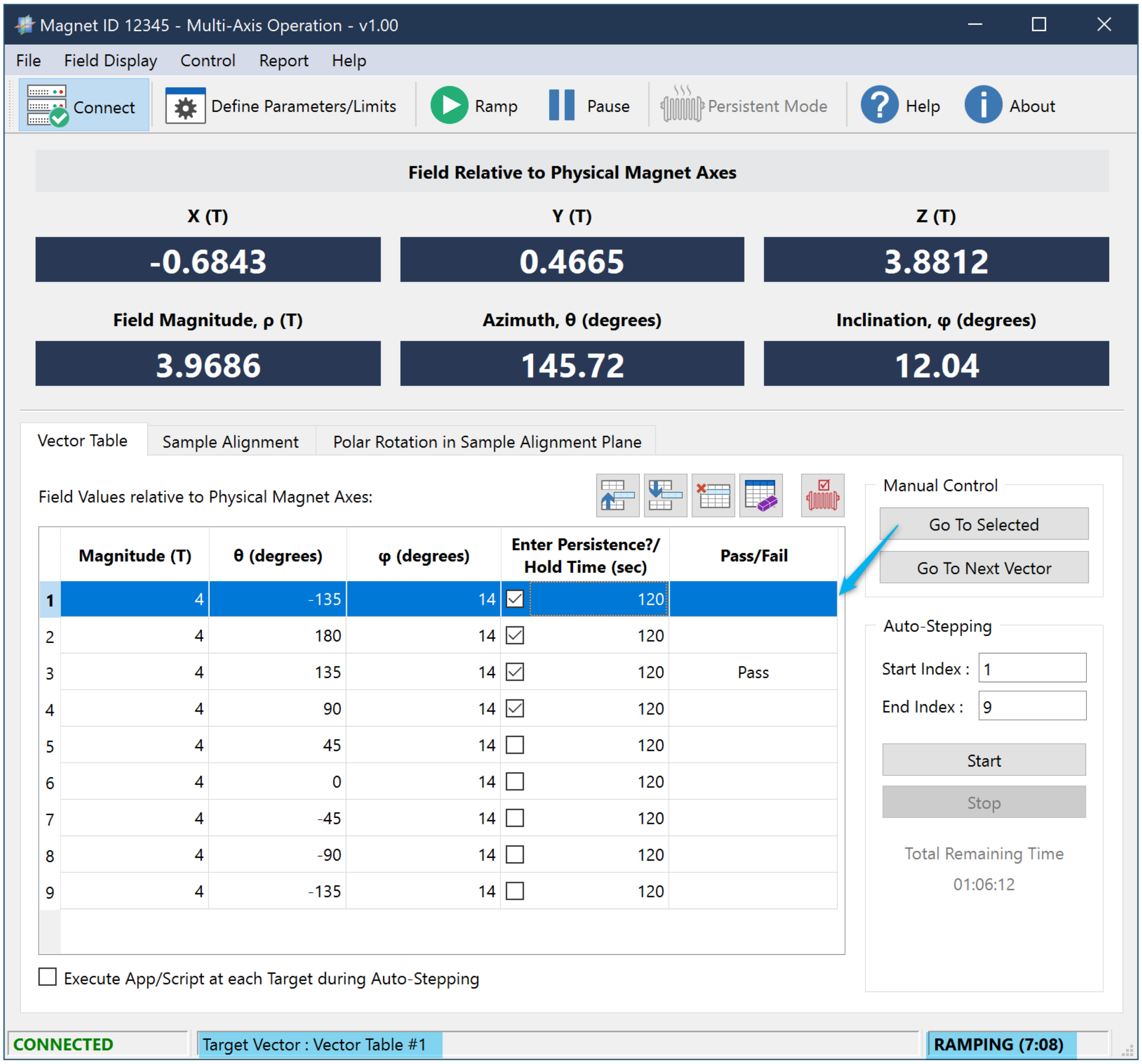


## Step 5: Manually select or Autostep the vectors in the table

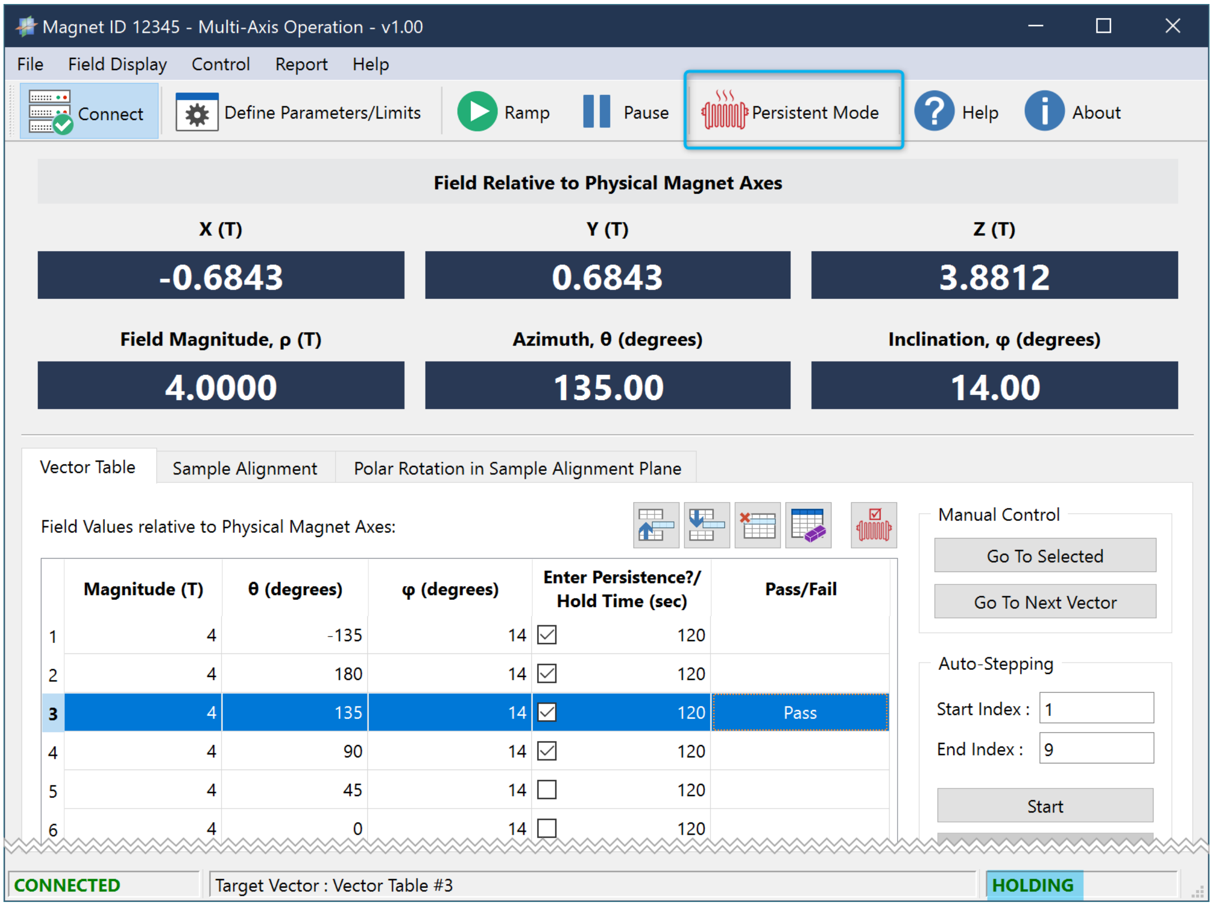
Once the vectors are loaded in the *Vector Table* tab, use the Manual (illustrated below with "Go To Selected") or Auto-Stepping controls to execute the vectors. The system status bar will show a **RAMPING** indication while the field is changing to match the selected target field vector (the value in parenthesis is an *estimate* of the time remaining until the target field is reached).

Auto-stepping will step through the vector table and hold at each vector for the dwell time specified in the “Time (sec)” column of the table.

Each vector visited successfully will get a “Pass” indication in the vector table. Quenches will generate a “Fail” indication and the quench currents at time of failure will be posted in the vector table.




When a vector is achieved, the rightmost status area at the bottom of the window will change to **HOLDING** (as shown below). This indicates all axes have reached the necessary field for the selected (target) vector. Manually selected vectors will maintain the holding status indefinitely as long as the system continues operating. With magnets that have persistent switches, the operator may opt to place the magnet in persistent mode.

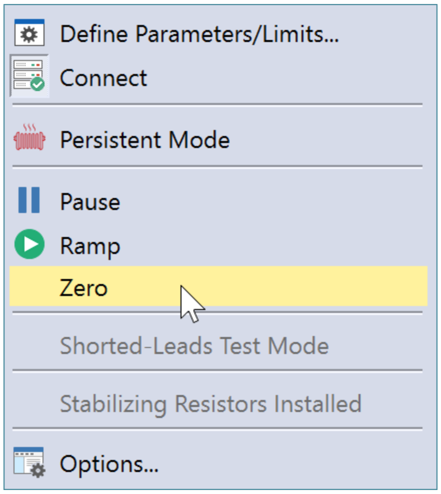



## Step 6: Save the results

Use the *Report | Generate Excel Report...* menu item to export all magnet parameters and the vector table results to an Excel .xlsx file (compatible with Office 2007 or later).

> **NOTE:** Use the *File | Save Settings...* menu item to save the *complete state of the application* for restoration in the future without having to reenter any of the magnet parameters or reload the tables. Use *File | Load Settings...* to reload the saved settings for subsequent sessions.

## Step 7: Return the magnetic field to zero if desired




Use the *Control | Zero* menu item to return the system to zero field at the end of a session if desired.

If the magnet is in persistent mode (i.e. switches are cooled), then the Zero function commands the power supply current outputs to zero but the magnet current and internal field is preserved. With the magnet in persistent mode, the power supply systems can be powered off.

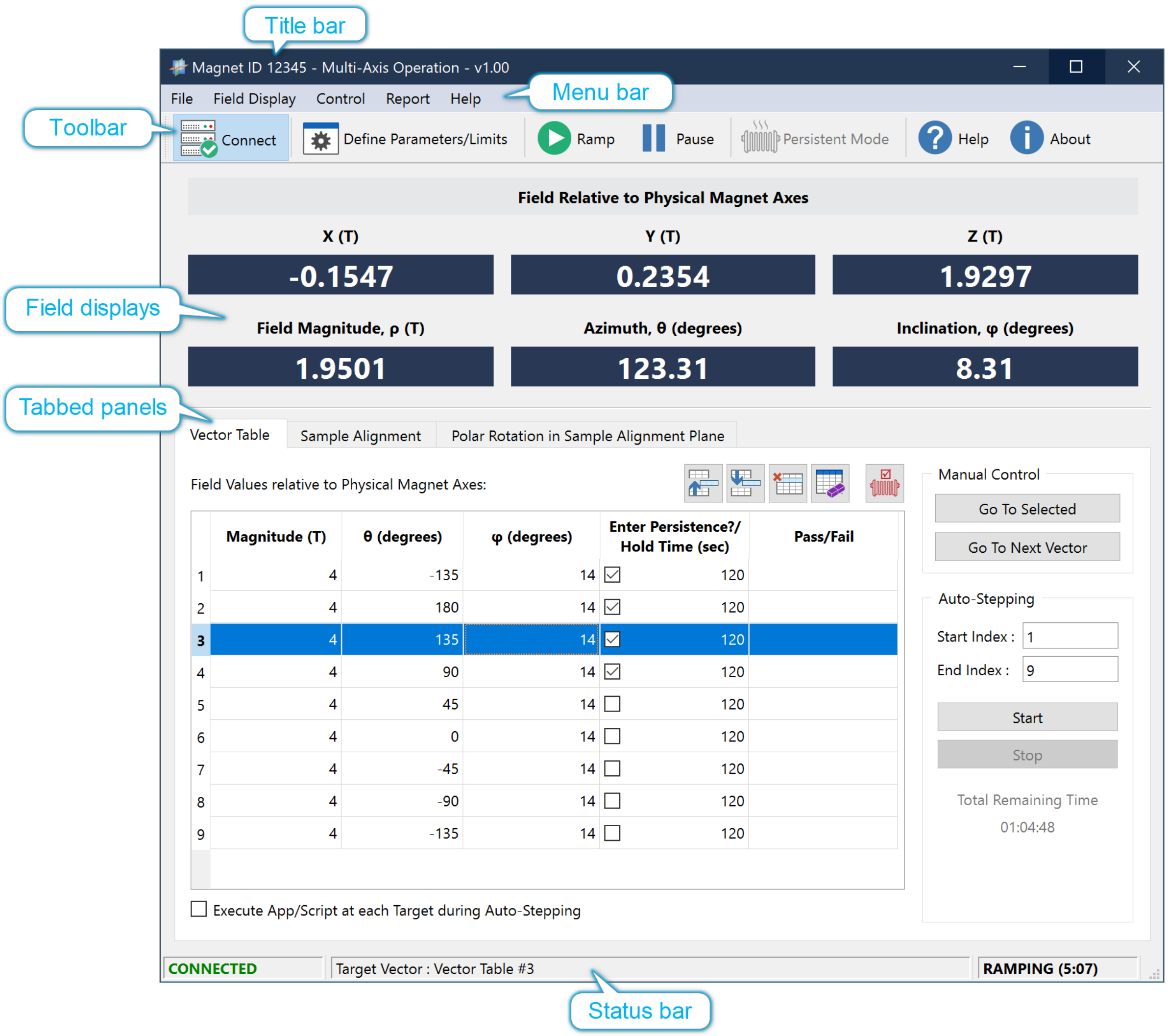
 

# Feature Descriptions

# 2 Feature Descriptions

## 2.1 Main Window Layout

The Main Window of the Multi-Axis Operation application consists of a title bar, a menu bar, a toolbar, tabbed panels, and a status area.




### Title bar

The title bar shows the application name and version, along with the present Magnet ID as defined in the *Define Parameters/Limits* dialog.

### Menu bar

The menu bar contains selections for numerous actions such as file saving and loading, display and units preferences, connection and control, and reporting. See the Menu Bar Items topic for more details.

### Toolbar

The toolbar contains common actions such as connection control and magnet configuration. Items are enabled and disabled as appropriate during the system operation. See the Toolbar Items topic for more details.

### Tabbed panels

The tabbed area of the main window includes the following views:

| | |
|---|---|
| **Vector Table** | A spreadsheet-like view of a list of field vectors (all relative to the physical coil axes of the magnet) as defined by the operator. |
| **Sample Alignment** | Entry of the sample alignment vectors which can be locked. |
| **Polar Rotation in Sample Alignment Plane** | Counterclockwise rotation in polar coordinates in a plane aligned with the sample alignment vectors. |

> **NOTE:** The panel tabs can be dragged to reorder them per your preference. This new order is saved and restored for subsequent application launches.

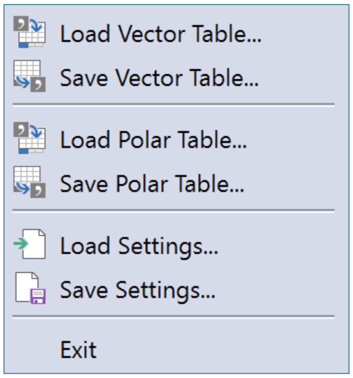
### Status bar

The Status bar contains three sub-panels. From left-to-right, the panels include the connection status, general error and status messages, and the power supply system status. Critical errors will be highlighted in red text.

## 2.2 Menu Bar Items

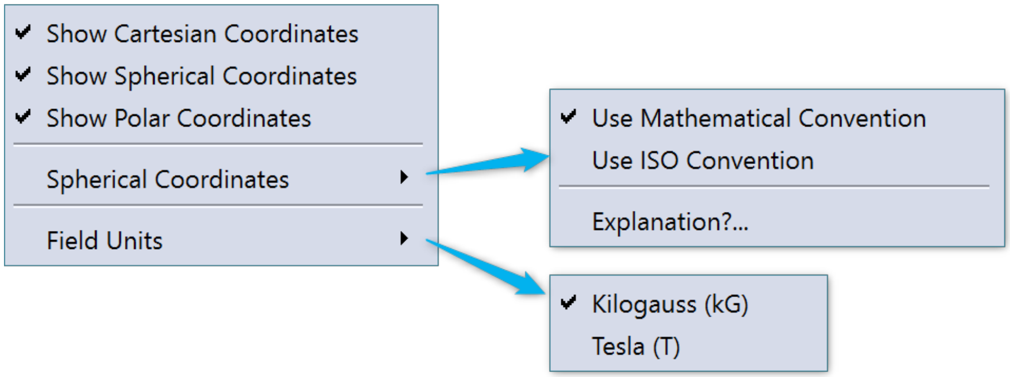

Each menu and contained item is described in the following paragraphs.

## File Menu



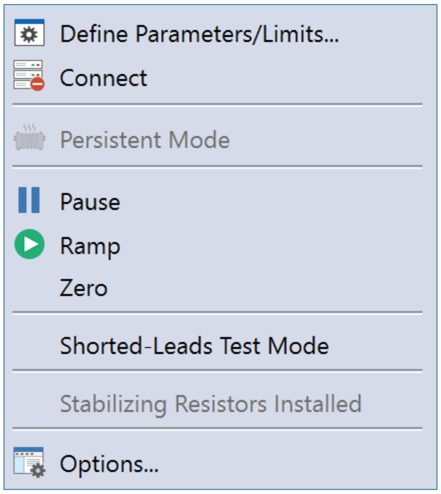

| | |
|---|---|
| **Load Vector Table...** | Displays a file chooser dialog to select a vector table file in Excel-compatible, comma-separated values (CSV) format. |
| **Save Vector Table...** | Displays a file chooser dialog to select a file for saving the present vector table in Excel-compatible, comma-separated values (CSV) format. |
| **Load Polar Table...** | Displays a file chooser dialog to select a polar table in Excel-compatible, comma-separated values (CSV) format. |
| **Save Polar Table...** | Displays a file chooser dialog to select a file for saving the present polar table in Excel-compatible, comma-separated values (CSV) format. |
| **Load Settings...** | Displays a file chooser dialog to select a previously saved settings (.sav extension) file. This file contains and restores the *complete state* of the Multi-Axis Operation application. *This menu is only active in the disconnected state – reloading all the settings is not allowed while a system is actively connected.* |
| **Save Settings...** | Displays a file chooser dialog to select a file for saving the *complete application state* including all the magnet parameters/limits settings and table values with the .sav extension in a format specific to the application. |
| **Exit** | Closes all system connections and exits the application. The present state of the application interface is cached and will be restored on the next launch, except for the vector and polar table values. |

## Field Display Menu




| | |
|---|---|
| **Show Cartesian Coordinates** | If checked, the main window will show the field in cartesian coordinates relative to the physical coil axes of the magnet in the selected field units. |
| **Show Spherical Coordinates** | If checked, the main window will show the field in spherical coordinates relative to the physical coil axes of the magnet, including the magnitude in the selected field units and the azimuth and inclination angles in *degrees* in the selected coordinate convention. **The magnitude must be a positive value and the inclination angle from the positive Z axis must be ≥ 0 and ≤ 180 degrees.** |
| **Show Polar Coordinates** | If checked, the main window will show the projection of the present magnet field measurement onto the Sample Alignment Plane in polar coordinates relative to the Alignment Vector #1. The fields will not show a value if an alignment plane is not defined. |
| **Spherical Coordinates »** | Chooses the convention of the θ and φ labels. Choose the *Explanation?* menu item to visit a Wikipedia page that provides the definition of the two conventions. |
| **Field Units »** | Chooses either kilogauss (kG) or tesla (T) units for field. This setting may only be changed while in the disconnected state. Once a connection is active, the field units choice is locked for that connection session. |

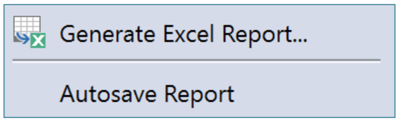
## Control Menu




| | |
|---|---|
| **Define Parameters/ Limits...** | Displays the magnet parameters and limits dialog (same function as toolbar item). |
| **Connect** | Initiates or closes a connection to a magnet system and its associated Model 430 Programmers (same function as toolbar item). |
| **Persistent Mode** | Enters and exits persistent mode if a persistent switch is installed on one or more magnet axes (otherwise the menu item is disabled). If blue with a lock shown on the icon, then one or more magnet axes are in persistent mode (heater is OFF). You must first exit persistent mode (i.e. select the menu item to toggle the heater) to change the target field. |
| **Pause** | Places the magnet system in a paused state (all Model 430 Programmers are paused). System status is displayed as **PAUSED**. |
| **Ramp** | Places the magnet system in a ramping state to the active vector. System status is displayed as **RAMPING** until the vector is achieved at which time the status becomes **HOLDING**. |
| **Zero** | Places the magnet system in a zeroing state where the field is ramped to zero without changing any Model 430 parameters. The system status will change to **AT ZERO** when zero field is achieved. |
| **Shorted-Leads Test Mode** | A special test mode that *must be selected before a Connect*. This allows a magnet system's power supplies and Model 430 Programmers to be tested on shorted power leads before actually |

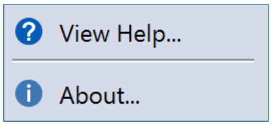
| | |
|---|---|
| | operating the magnet. This setting is *not* saved or restored between sessions, and the default is inactive.<br><br>The test procedure is as follows:<br>1. With the power supply systems OFF (supplies and Model 430's) with the magnet at zero field, disconnect the power leads from the magnet lead terminals and short them together at the cryostat.<br>2. Power on the power supply systems.<br>3. Choose this menu item so that it is checked.<br>4. *Connect* and load a vector table to test.<br>5. After testing is complete, set the app state to disconnected.<br>6. Ensure the power supply systems are OFF and reconnect the power leads to the magnet at the cryostat.<br>7. Uncheck this menu item.<br>8. Power the power supply systems ON and *Connect* to operate the magnet. |
| **Stabilizing Resistors Installed** | Choose this menu item so that it is checked if optional stabilizing resistors are installed on every magnet axis which does not have a persistent switch. All non-switched axes must have the stabilizing resistor installed if this option is chosen. This option must be specified before connecting. |
| **Options...** | Global application options, see the Options topic page for more detail. |

### Report Menu



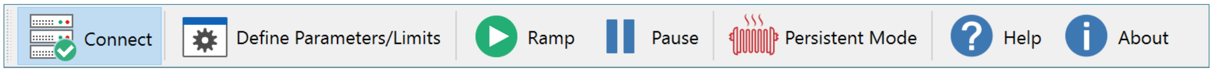

| | |
|---|---|
| **Generate Excel Report...** | Displays a file chooser dialog which prompts for a filename to which to save an Excel-formatted (.xlsx) file (Excel 2007 or later) with the magnet parameters/limits and vector table results. |
| **Autosave Report** | If checked, the application will attempt to autosave a report when an Autostep session completes (either successfully or unsuccessfully) to the same folder from which the last *Load Table Vectors...* file was loaded. Multiple saves will contain an incremented integer value in the filename to avoid overwriting previous autosaves. |

### Help Menu




| | |
|---|---|
| **View Help...** | Displays this web-formatted Help in the default system web browser. |
| **About...** | Displays an information dialog incuding the license terms. |

## 2.3 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 multi-axis magnet system and Model 430 Programmers. Up to three (3) axes are supported. When highlighted (as shown above with a light blue background color), the connection is *active*. Press again while highlighted to close the connection. |

| | |
|---|---|
| **Define Parameters/Limits** | Displays the magnet parameters and limits dialog to provide the necessary information to operate the magnet system within manufacturer-prescribed limits. |
| **Ramp** | Shortcut for the Control \| Ramp menu item:<br>Places the magnet system in a ramping state to the active vector. System status is displayed as **RAMPING** until the vector is achieved at which time the status becomes **HOLDING**. |
| **Pause** | Shortcut for the Control \| Pause menu item:<br>Places the magnet system in a paused state (all Model 430 Programmers are paused). System status is displayed as **PAUSED**. |
| **Persistent Mode** | Shortcut for the Control \| Persistent Mode menu item:<br>Enters and exits persistent mode if a persistent switch is installed on one or more magnet axes (otherwise the menu item is disabled). If the icon is a blue coil with a lock symbol, then one or more magnet axes are in persistent mode (heater is OFF). You must first exit persistent mode (i.e. select the menu item to toggle the heater) to change the target field. |
| **Help** | Displays this Help file in the default system web browser application (Javascript support required). |
| **About** | Displays an information and license terms dialog. |

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

## 2.4 Vector Table Tab

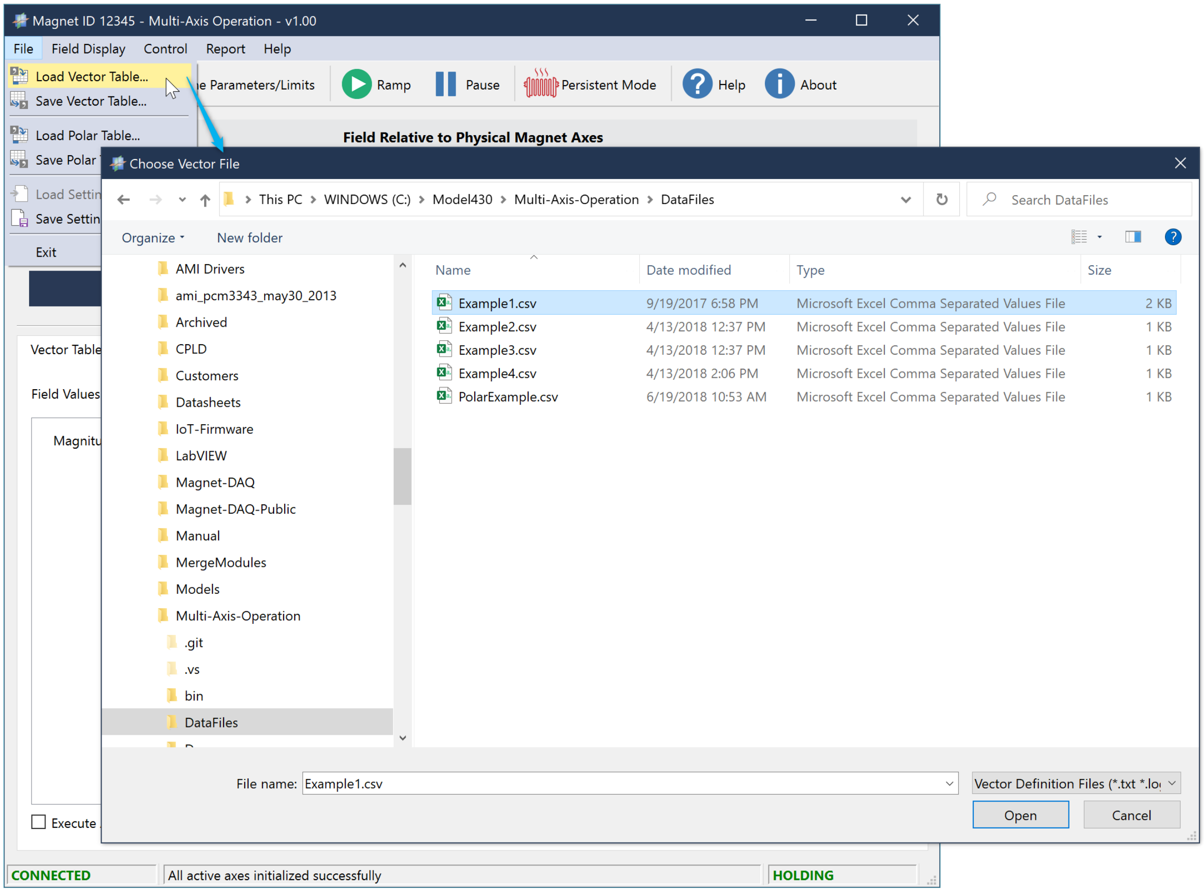
The *Vector Table* Tab allows the operator to load, edit, and save a list of vectors in spherical or Cartesian coordinates *relative to the physical axes of the magnet*. The file format is the standard comma-separated values (CSV) format. The specific format recognized by the Multi-Axis Operation app is defined in detail in the Vector Table Format topic.

*Table edit icons*

> **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 above) to add a row above or below the selected row, delete a row, or clear the entire table, respectively. More advanced editing of a vector table is better performed using Microsoft Excel (or other compatible application) that can load, edit, and save comma-separated values (CSV) files.

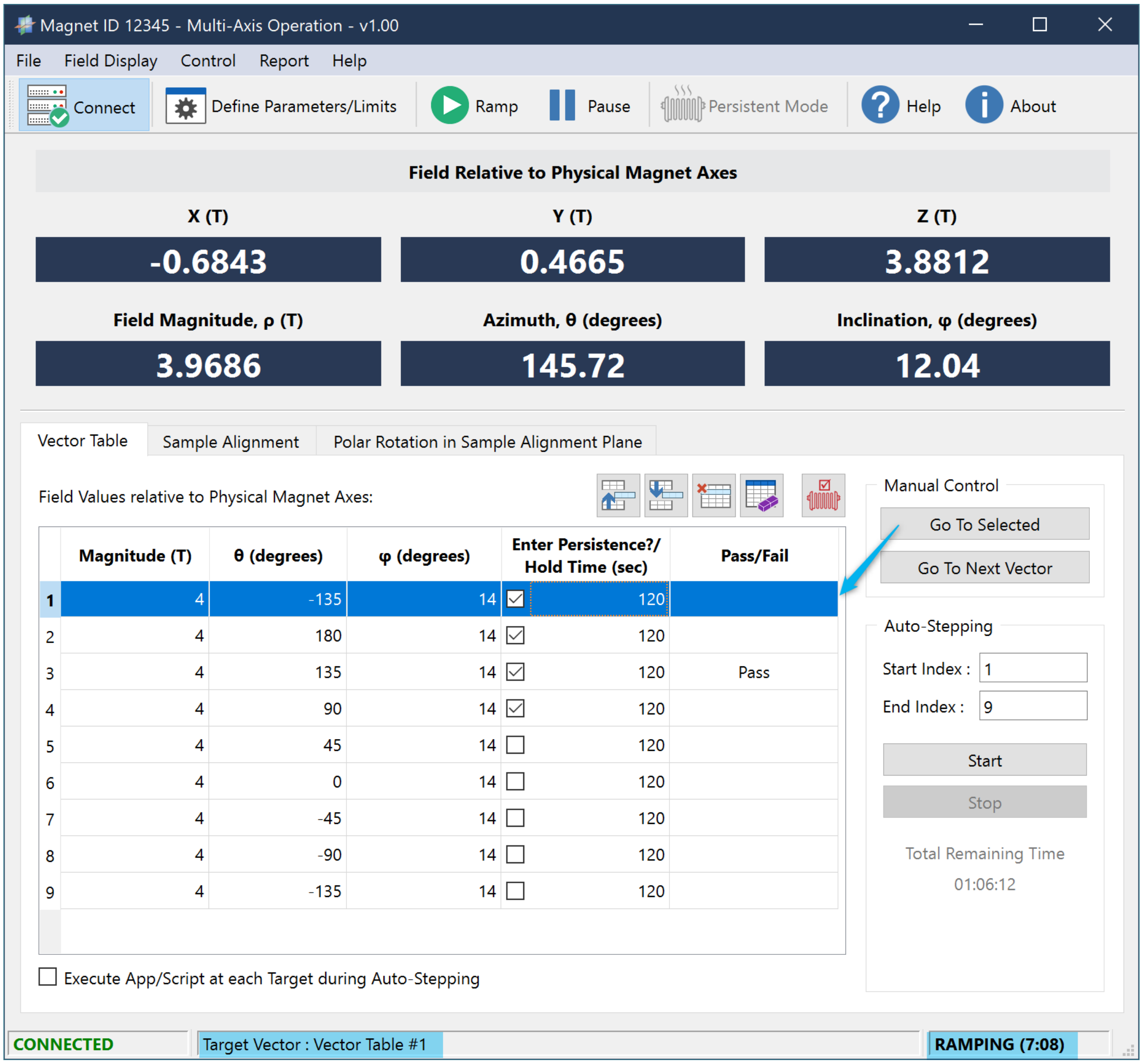
### Importing Vector Table Data

Selecting *File | Load Vector Table...* menu item displays a file browser to locate a vector table file to load in the table as shown in the illustration below. Any prior values in the table are discarded.




Once the vector table is loaded, you may manually choose a vector by clicking on a line in the table so that it is highlighted in blue. You may then press the *Go To Selected* button to command the system to ramp to the vector.

**CAUTION:** The *Magnet Parameters/Limits* values are used to determine the fastest ramping profile available to realize the vector and yet remain within the prescribed limits of magnet operation. It is critical that these parameters and limits be entered correctly according to the manufacturer's specifications. If a vector is commanded that is outside of the safe limits, the app will announce an error and ignore the entry.




**NOTE:** If the magnet for which the application is configured has one or more persistent switches, then each target row includes an **Enter Persistence?** checkbox. If the box is checked, then during *Auto-Stepping* the system will automatically enter persistent mode (by cooling all switches) during the **Hold Time**, and then exit persistence (by heating all switches) at that target field or current. Ensure the heating and cooling times for the persistent switch for each axis are set according to the magnet manufacturer's recommendations.

The "heater coil" checkmark tool button above the table will toggle the **Enter Persistence?** checkbox states for *all the rows* in the table. The default checkbox state on vector table loads can be set in the Options dialog. The *Save Settings.../Load Settings...* function saves and restores *all* manually chosen **Enter Persistence?** checkbox states.

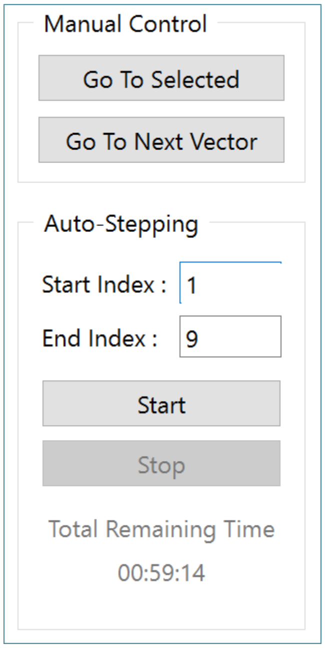
## Two important terms are used in reference to the table: *Target* and *Auto-Stepping*

The **Target Vector** is defined as the field vector that is presently programmed in the magnet system; or in other words, the Target Setpoint and the ramp rate values for each Model 430 controlling each axis that have been remotely programmed to realize the vector. The Target Vector is highlighted in the table when it is successfully made the target. If a vector reaches the **HOLDING** state, then it is marked as "Pass" in the Pass/Fail column.

> **NOTE:** The application calculates and sets the ramping parameters so that all axes arrive at the required field value at *approximately* the same time, unless an axis is already at the necessary field value to realize the Target Vector.

In addition to manual controls which will hold the Target Vector indefinitely, the table also has an **Auto-Stepping** sequencing feature whereby the field vectors can be executed unattended 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 vector is realized.

The right-side controls are described below:



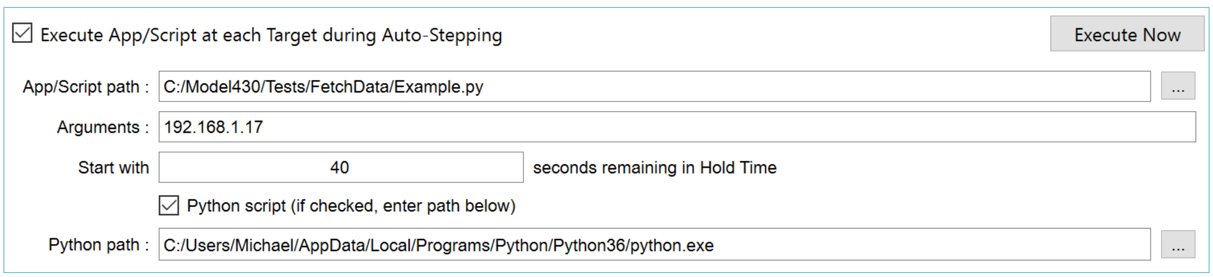

| | |
|---|---|
| **Go To Selected** | Makes the presently selected row the Target Vector and commands the system to ramp to the vector. |
| **Go To Next Vector** | Makes the next row in the table the Target Vector and commands the system to ramp to the vector. |
| **Start Index** | Specifies the starting 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. |

## 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 in the above example 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 provided for the Python script to consume, and not the Python executable.

## 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. Please note these values are referenced to the physical axes of the magnet.

The Example.py script is provided in the Advanced Features section of this Help illustrating the use of the special variables as Arguments.

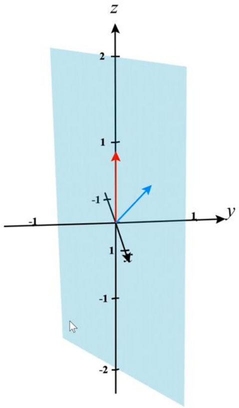
| Variable Synonyms | Description |
|---|---|
| **%MAGNITUDE%**<br>**%MAG%**<br>**$MAGNITUDE**<br>**$MAG** | Inserts the magnitude of the present magnet field in the selected field units. |
| **%AZIMUTH%**<br>**%AZ%**<br>**$AZIMUTH**<br>**$AZ** | Inserts the azimuth angle of the present magnet field in degrees. |
| **%INCLINATION%**<br>**%INC%**<br>**$INCLINATION**<br>**$INC** | Inserts the inclination angle of the present magnet field in degrees. |
| **%FIELDX%**<br>**$FIELDX** | Inserts the X-axis component of the present magnet field in the selected field units. |
| **%FIELDY%**<br>**$FIELDY** | Inserts the Y-axis component of the present magnet field in the selected field units. |
| **%FIELDZ%**<br>**$FIELDZ** | Inserts the magnitude of the present magnet field in the selected field units. |
| **%TARG:MAG%**<br>**$TARG:MAG** | Inserts the magnitude of the target magnet field in the selected field units.<br><br>The present magnitude and the target magnitude should match when the HOLDING state is reached. |
| **%TARG:AZ%**<br>**$TARG:AZ** | Inserts the target azimuth angle in degrees. |

| | |
|---|---|
| | The present azimuth angle and the target azimuth angle should match when the HOLDING state is reached. |
| **%TARG:INC%**<br>**$TARG:INC** | Inserts the target inclination angle in degrees.<br><br>The present inclination angle and the target inclination angle should match when the HOLDING state is reached. |
| **%TARG:X%**<br>**$TARG:X** | Inserts the X-axis component of the target field in the selected field units.<br><br>The present X-axis field and the target X-axis field should match when the HOLDING state is reached. |
| **%TARG:Y%**<br>**$TARG:Y** | Inserts the Y-axis component of the target field in the selected field units.<br><br>The present Y-axis field and the target Y-axis field should match when the HOLDING state is reached. |
| **%TARG:Z%**<br>**$TARG:Z** | Inserts the magnitude of the target field in the selected field units.<br><br>The present Z-axis field and the target Z-axis field should match when the HOLDING state is reached. |

## 2.5 Sample Alignment Tab

The Sample Alignment Tab provides an interface for specification of two sample alignment vectors, or a vector and a point if you wish. The two together define a *sample alignment plane* as defined by the normal vector to the plane formed by the cross-product of the two alignment vectors.

The video at right illustrates the example alignment vectors as shown in the main window image below. The red vector is Sample Alignment Vector #1 and the blue vector is the Sample Alignment Vector # 2. The light blue surface is the plane of rotation defined by the two alignment vectors.



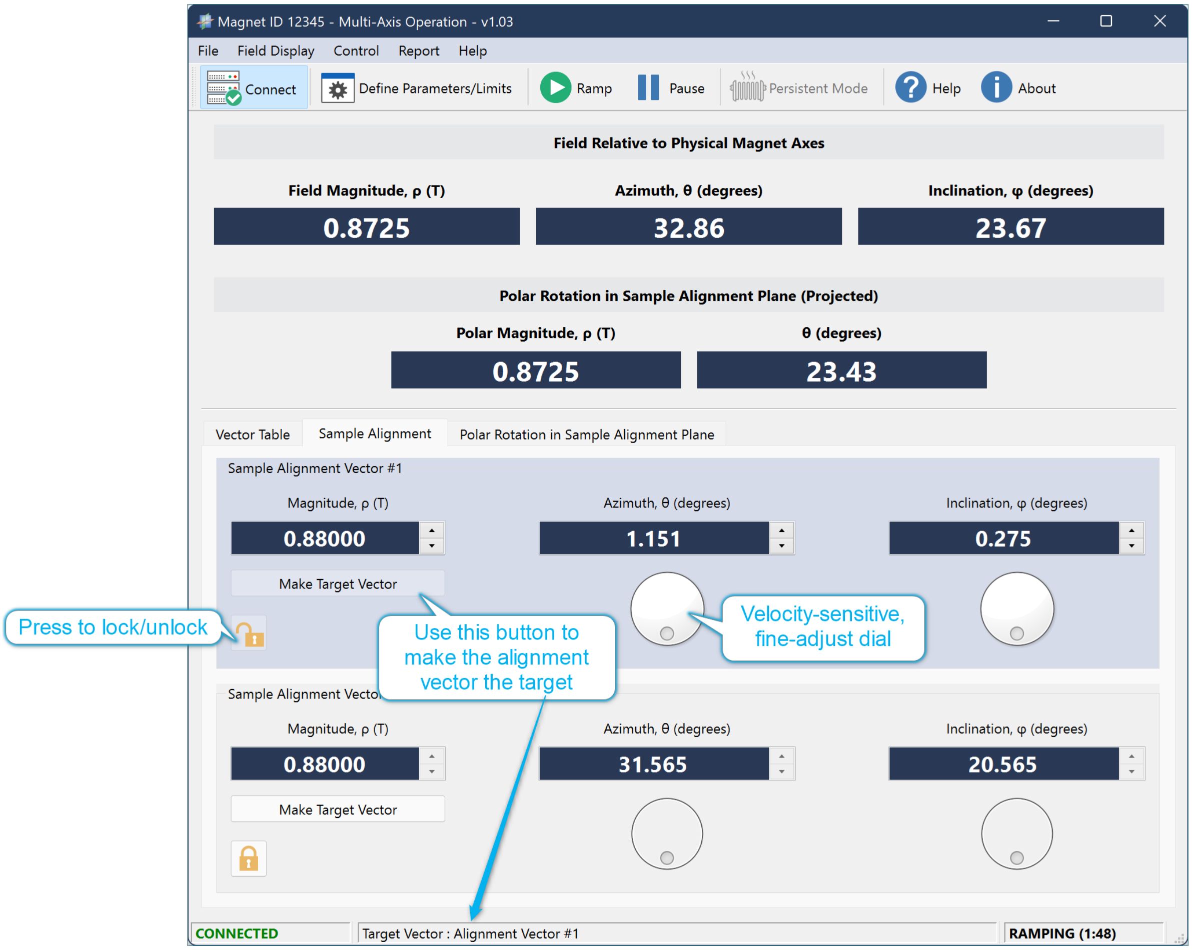

*Visualization of Example Sample Alignment Plane*

The interface provides a lock feature which prevents accidental changes in the vectors. To adjust each, first unlock each vector by pressing the lock icon to toggle it unlocked (shown at left). Once unlocked, you may type new values into each field, use the up/down arrows for each, or for the angles use the velocity-sensitive, fine adjust dials.

Use the "Make Target Vector" button to choose an align vector as the Target Vector. The system will immediately begin ramping to the vector, hold when realized, and will immediate try to realize any adjustments in the vector parameters as you adjust them.

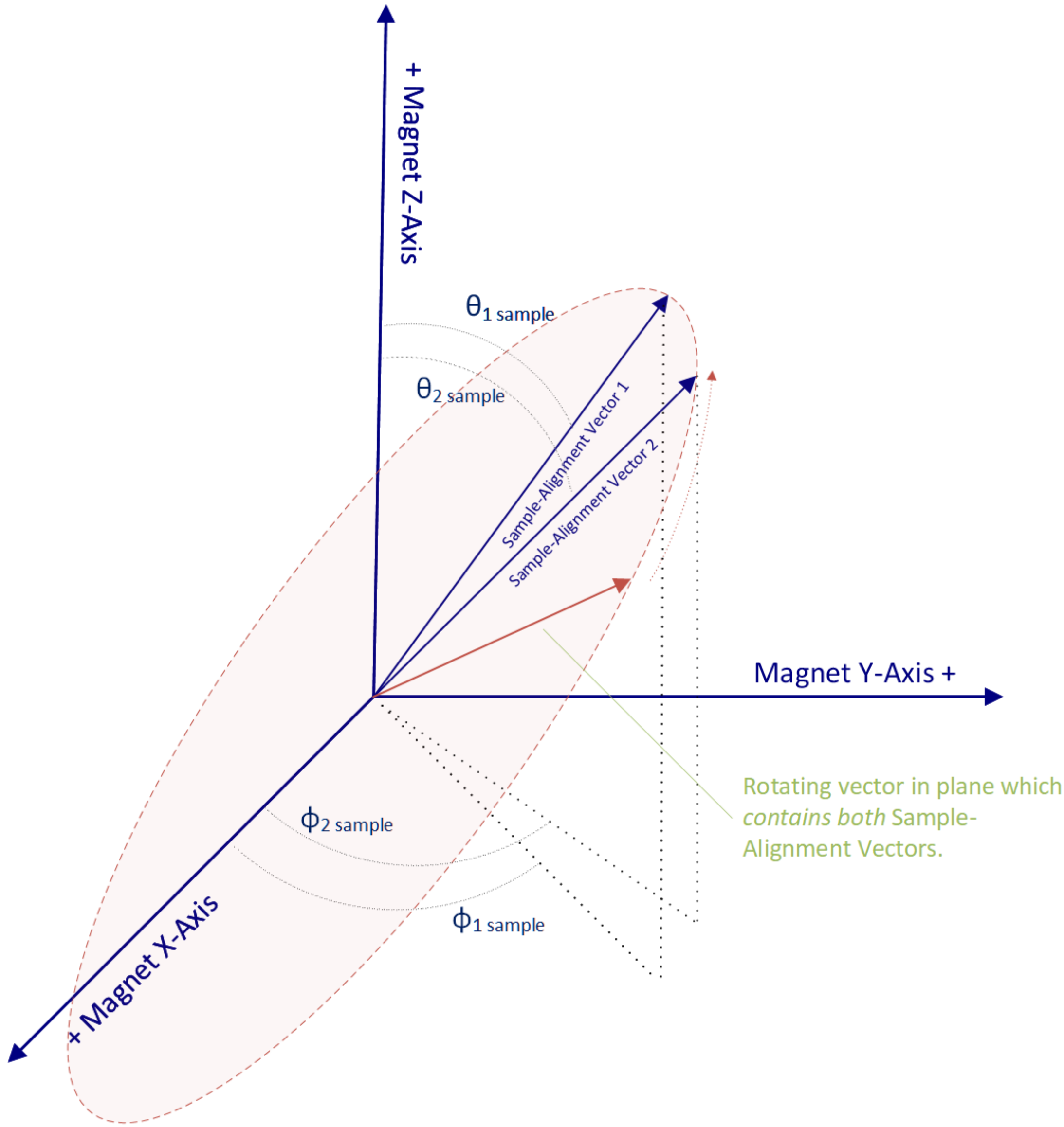
The image below shows the Sample Alignment Vector #1 unlocked and selected as the Target Vector. The status bar in the illustration below shows the system ramping to the vector with approximately 1:48 minutes of ramping time remaining to realize the vector.

You may adjust the vector components by direct numerical entry in the fields, use of the up/down arrows for the field, or by adjustment of the dials for the azimuth and inclination angles.



Magnet ID 12345 - Multi-Axis Operation - v1.03
File Field Display Control Report Help
Connect
Define Parameters/Limits
Ramp
Pause
Persistent Mode
Help
About
Field Relative to Physical Magnet Axes
Field Magnitude, ρ (T)
Azimuth, θ (degrees)
Inclination, φ (degrees)
0.8725
32.86
23.67
Polar Rotation in Sample Alignment Plane (Projected)
Polar Magnitude, ρ (T)
θ (degrees)
0.8725
23.43
Vector Table
Sample Alignment
Polar Rotation in Sample Alignment Plane
Sample Alignment Vector #1
Magnitude, ρ (T)
Azimuth, θ (degrees)
Inclination, φ (degrees)
0.88000
1.151
0.275
Make Target Vector
Press to lock/unlock
Use this button to make the alignment vector the target
Velocity-sensitive, fine-adjust dial
Magnitude, ρ (T)
Azimuth, θ (degrees)
Inclination, φ (degrees)
0.88000
31.565
20.565
Make Target Vector
CONNECTED
Target Vector : Alignment Vector #1
RAMPING (1:48)

## 2.6 Polar Rotation in Sample Alignment Plane Tab




The Polar Rotation in Sample Alignment Plane Tab allows the operator to load, edit, and save a list of vectors in polar coordinates relative to the Sample Alignment Vector #1 and rotated in the sample alignment plane. The positive (+) angle of rotation is the right hand rule direction for the normal vector defined by the cross-product of the two alignment vectors ($v_1 \times v_2$). The rotation angle is referenced to the Sample Alignment Vector #1.

A visualization of this relationship is illustrated above. The maroon-colored vector is a vector in the rotation plane (pink) defined by the two sample alignment vectors.

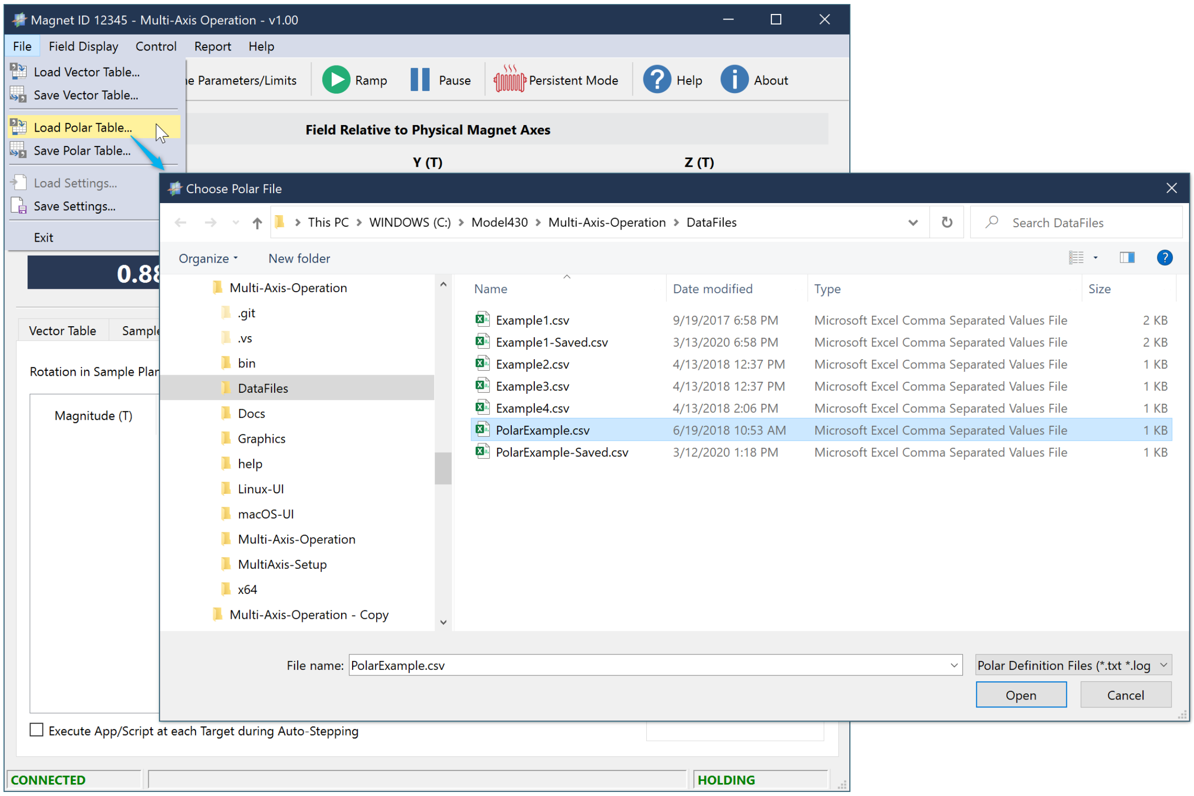
*Table edit icons*

**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 above) 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.

## Importing Polar Table Data

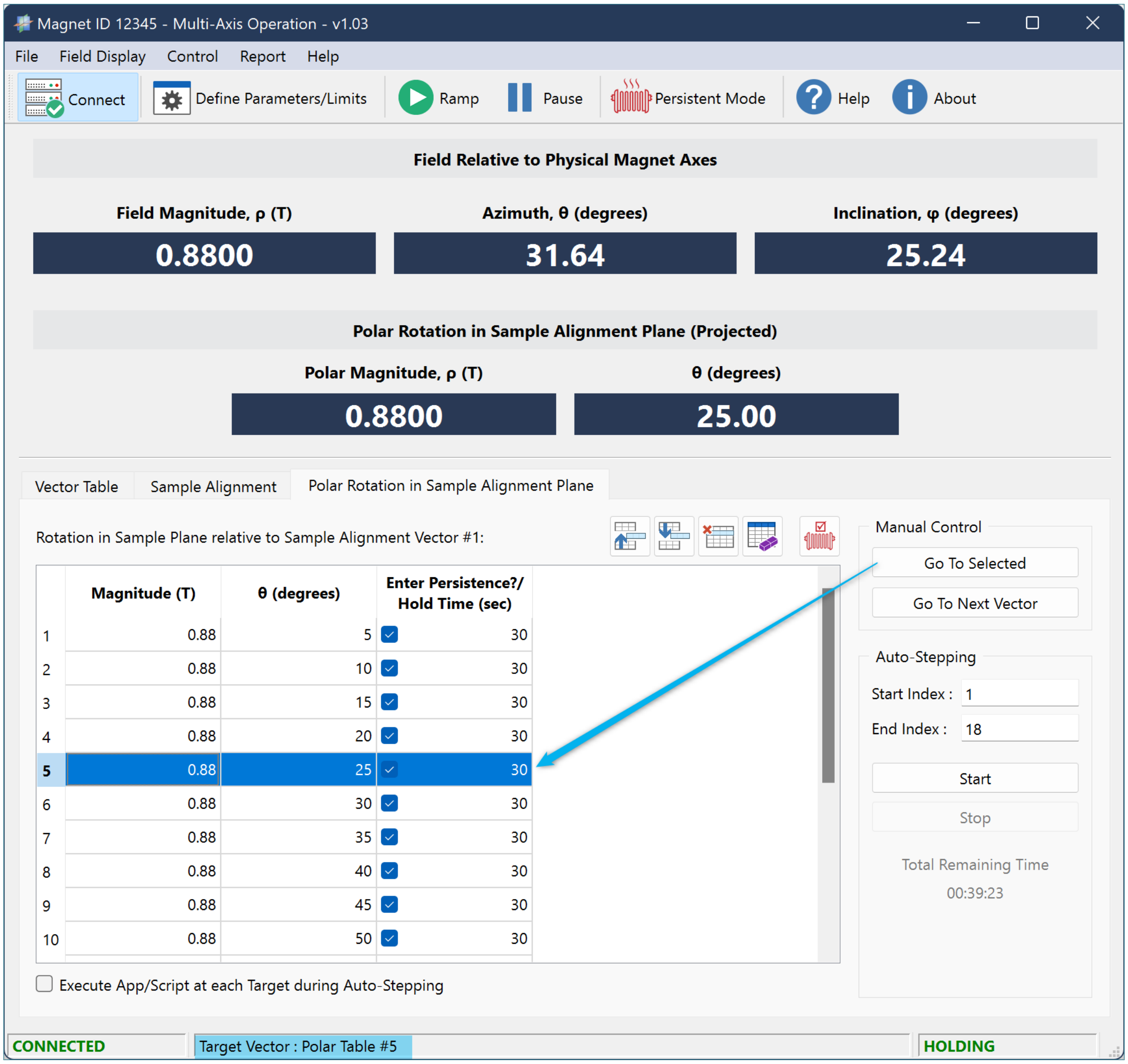
Selecting *File | Load Polar Table...* menu item displays a file browser to locate a polar table file to load in the table as shown in the illustration below. Any prior values in the table are discarded.

The table file format is the standard comma-separated values (CSV) format. The specific format recognized by the Multi-Axis Operation app is defined in detail in the Polar Table Format topic.




Once the polar table is loaded, you may manually choose a vector by clicking on a line in the table so that it is highlighted in blue. You may then press the *Go To Selected* button to command the system to ramp to the vector.

**CAUTION:** The *Magnet Parameters/Limits* values are used to determine the fastest ramping profile available to realize the vector and yet remain within the prescribed limits of magnet operation. It is critical that these parameters and limits be entered correctly according to the manufacturer's specifications. If a vector is commanded that is outside of the safe limits, the app will announce an error and ignore the entry.




**NOTE:** If the magnet for which the application is configured has one or more persistent switches, then each target row includes an **Enter Persistence?** checkbox. If the box is checked, then during *Auto-Stepping* the system will automatically enter persistent mode (by cooling all switches) during the **Hold Time**, and then exit persistence (by heating switches) at that target field or current. Ensure the heating and cooling times for the persistent switch for each axis are set according to the magnet manufacturer's recommendations.

The "heater coil" checkmark tool button  above the table will toggle the **Enter Persistence?** checkbox states for *all the rows* in the table. The default checkbox state on polar table loads can be set in the Options dialog. The *Save Settings.../Load Settings...* function saves and restores *all* manually chosen **Enter Persistence?** checkbox states.

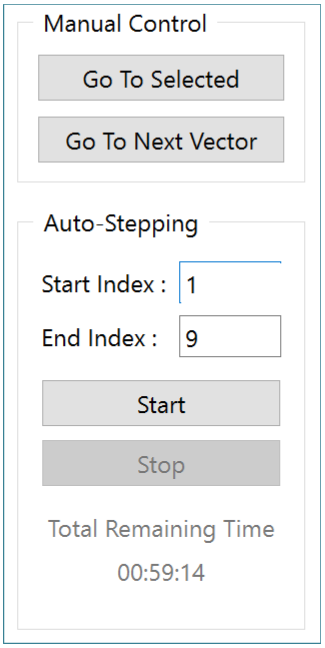
**Two important terms are used in reference to the tables: *Target Vector* and *Auto-Stepping***

The **Target Vector** is defined as the field vector that is presently programmed in the magnet system; or in other words, the Target Setpoint and the ramp rate values for each Model 430 controlling each axis that have been remotely programmed to realize the vector. The Target Vector is highlighted in the table when it is successfully made the target. Once the system reaches the target field it changes to the **HOLDING** state.

> **NOTE:** The application calculates and sets the ramping parameters so that all axes arrive at the required polar field value at *approximately* the same time, unless an axis is already at the necessary field value to realize the Target Vector.

In addition to manual controls which will hold the Target Vector indefinitely, the table also has an **Auto-Stepping** sequencing feature whereby the polar field vectors can be executed unattended 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 polar vector is realized.

The right-side controls are described below:



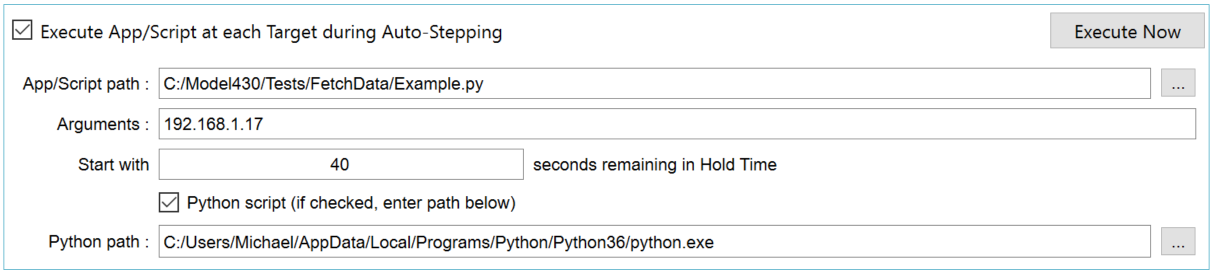

| | |
|---|---|
| **Go To Selected** | Makes the presently selected row the Target Vector and commands the system to ramp to the vector. |
| **Go To Next Vector** | Makes the next row in the table the Target Vector and commands the system to ramp to the vector. |
| **Start Index** | Specifies the starting 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. |

## 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 in the above example 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 provided for the Python script to consume, and not the Python executable.

## 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. Please note the values not containing the **POLAR** keyword are referenced to the *physical axes* of the magnet.

The Example.py script is provided in the Advanced Features section of this Help illustrating the use of the special variables as Arguments.

| Variable Synonyms | Description |
|---|---|
| **%POLAR:MAG%**<br>**$POLAR:MAG** | Inserts the present magnet field magnitude projected into the sample alignment plane in the selected field units. |
| **%POLAR:ANGLE%**<br>**$POLAR:ANGLE** | Inserts the present magnet field angle projected into the sample alignment plane in degrees as referenced to the Sample Alignment Vector #1 using the right hand rule for the normal vector defined by the cross-product of the two alignment vectors ($v_1 \times v_2$). |
| **%POLAR:TARG:MAG%**<br>**$POLAR:TARG:MAG** | Inserts the target field magnitude in the sample alignment plane in the selected field units.<br><br>The present polar magnitude and the target polar magnitude should match when the HOLDING state is reached. |
| **%POLAR:TARG:ANGLE%**<br>**$POLAR:TARG:ANGLE** | Inserts the target field angle in the sample alignment plane in degrees, referenced to the Sample Alignment Vector #1 using the right hand rule for the normal vector defined by the cross-product of the two alignment vectors ($v_1 \times v_2$).<br><br>The present polar angle and the target polar angle should match when the HOLDING state is reached. |
| **%MAGNITUDE%**<br>**%MAG%**<br>**$MAGNITUDE**<br>**$MAG** | Inserts the magnitude of the present magnet field in the selected field units. |
| **%AZIMUTH%**<br>**%AZ%**<br>**$AZIMUTH** | Inserts the azimuth angle of the present magnet field in degrees. |

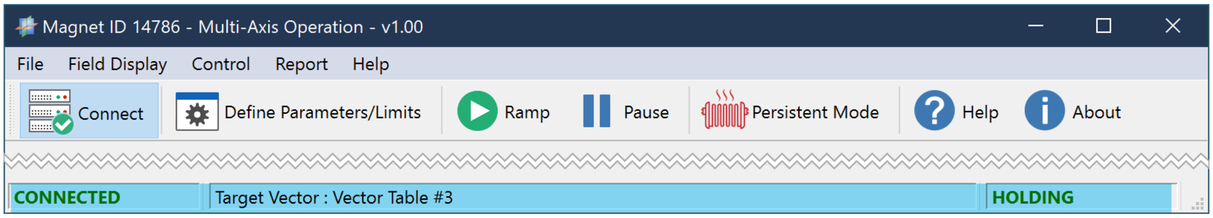
| | |
|---|---|
| **$AZ** | |
| **%INCLINATION%**<br>**%INC%**<br>**$INCLINATION**<br>**$INC** | Inserts the inclination angle of the present magnet field in degrees. |
| **%FIELDX%**<br>**$FIELDX** | Inserts the X-axis component of the present magnet field in the selected field units. |
| **%FIELDY%**<br>**$FIELDY** | Inserts the Y-axis component of the present magnet field in the selected field units. |
| **%FIELDZ%**<br>**$FIELDZ** | Inserts the magnitude of the present magnet field in the selected field units. |
| **%TARG:MAG%**<br>**$TARG:MAG** | Inserts the magnitude of the target magnet field in the selected field units.<br><br>The present magnitude and the target magnitude should match when the HOLDING state is reached. |
| **%TARG:AZ%**<br>**$TARG:AZ** | Inserts the target azimuth angle in degrees.<br><br>The present azimuth angle and the target azimuth angle should match when the HOLDING state is reached. |
| **%TARG:INC%**<br>**$TARG:INC** | Inserts the target inclination angle in degrees.<br><br>The present inclination angle and the target inclination angle should match when the HOLDING state is reached. |
| **%TARG:X%**<br>**$TARG:X** | Inserts the X-axis component of the target field in the selected field units.<br><br>The present X-axis field and the target X-axis field should match when the HOLDING state is reached. |
| **%TARG:Y%**<br>**$TARG:Y** | Inserts the Y-axis component of the target field in the selected field units.<br><br>The present Y-axis field and the target Y-axis field should match when the HOLDING state is reached. |
| **%TARG:Z%**<br>**$TARG:Z** | Inserts the magnitude of the target field in the selected field units. |

| | |
|---|---|
| | The present Z-axis field and the target Z-axis field should match when the HOLDING state is reached. |

## 2.7 Status Bar

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

- The leftmost pane indicates the connection status (e.g. **CONNECTED** or **DISCONNECTED**) of the multi-axis system.
- The middle pane provides application status (such as the Target Vector selection as illustrated below) or error information. Critical error information will be shown in **bold red text**.
- The rightmost pane provides power supply system status (e.g. **PAUSED**, **RAMPING**, **HOLDING**, **AT ZERO**, or **QUENCH!**).
- When in the **RAMPING** state, an estimate in seconds of the remaining time needed to reach the target is displayed in parentheses.



# Configuration Panels

# 3 Configuration Panels

## 3.1 Magnet Parameters/Limits

The Magnet Parameters and Limits dialog provides entries for values specified by the magnet manufacturer for safe operation within prescribed limits. It is important that these values and limits be accurate for proper multi-axis operation, and the manufacturer should provide this information as part of the magnet delivery.

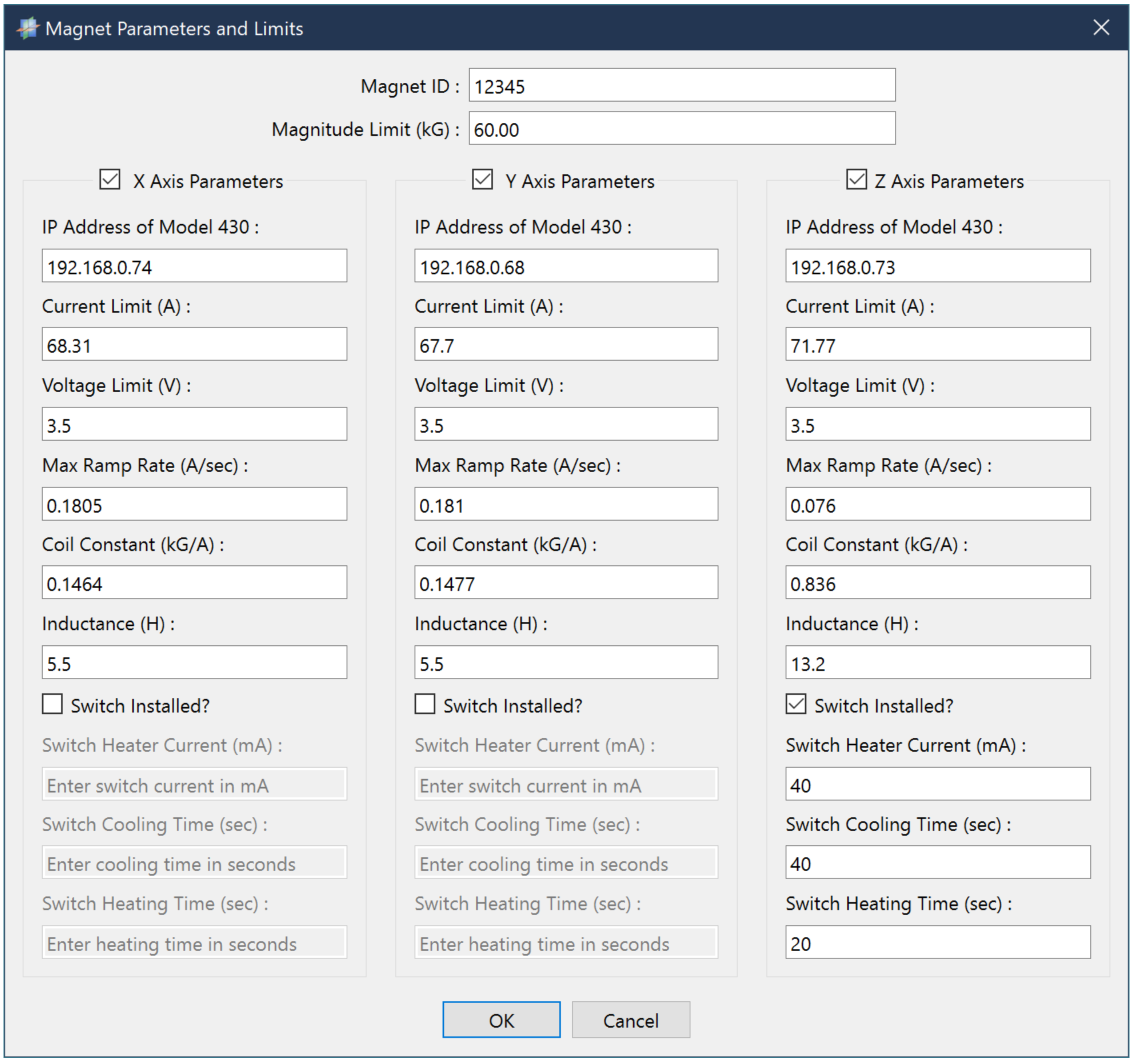
> **NOTE:** The entries in the dialog are cached between launches of the application, i.e. the last entered values are restored on launch. Use the *File | Load Settings...* menu item to reload previously saved parameters for a different magnet.

> **NOTE:** When changing the Field Units, the units in the dialog are changed and the field values are automatically converted to match the new preference.

The dialog is accessed using the *Define Parameters/Limits...* menu item or toolbar button and will appear as illustrated below. Use the *X-Axis Parameters*, *Y-Axis Parameters*, and *Z-Axis Parameters* checkboxes to indicate which axes are active for the magnet session. If an axis is unchecked, the parameters will be cleared for that axis and the axis will be disabled.

Once a system is in the **CONNECTED** state, the dialog values can only be *viewed* and cannot be modified (the dialog fields will be locked).

> **NOTE:** Disabled axes will limit which vector values can be realized in 3D coordinates. The application will announce an error and ignore a field vector entry if it cannot be achieved with the active axes.



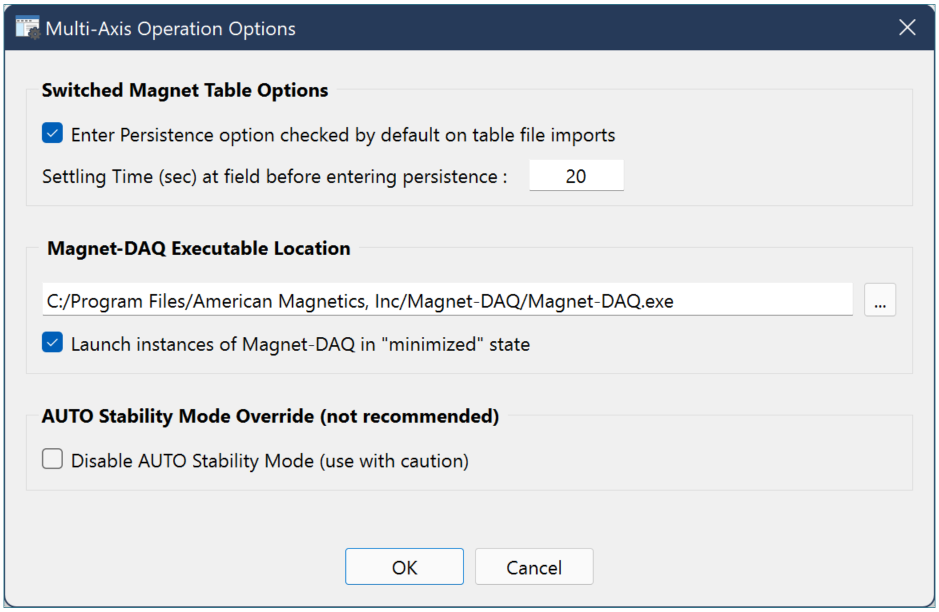

A detailed description of each parameter is provided in the following table:

| | |
|---|---|
| **Magnet ID** | An ID string for the magnet. AMI assigns unique Magnet IDs to all systems which also identifies the specifications and manufacturing record on file. |
| **Magnitude Limit (kG or T)** | The overall field magnitude limit for the magnet. |
| **IP Address of Model 430** | Specify the IP address of the Model 430 that is connected to the selected axis. AMI recommends static IP assignment for the Model 430 to prevent a DHCP server from changing the IP address between sessions. |
| **Current Limit (A)** | The current limit for the selected axis in amperes. |

| | |
|---|---|
| **Voltage Limit (V)** | The voltage limit for the selected axis in volts. This value should be the *Max Ramp Rate* × *Inductance* plus additional voltage to account for power lead resistance drop (typically 1 to 2V at the maximum current limit). The voltage limit is an additional safety factor to help prevent accidental activation of protection circuits internal to the magnet. |
| **Max Ramp Rate (A/sec)** | The maximum charge rate in amperes/second for the selected axis. |
| **Coil Constant (kG/A or T/A)** | The coil constant for the selected axis. The *Coil Constant* × *Current Limit* defines the maximum field allowed for the selected axis. |
| **Inductance (H)** | The inductance in henries of the selected axis. |
| **Switch Installed?** | A checkbox indicating whether a persistent switch is installed for the selected axis. If no switch is installed, the remaining switch-related entries will be disabled. |
| **Switch Heater Current (mA)** | If a persistent switch is installed for the selected axis, enter the heating current required in milliamperes. |
| **Switch Cooling Time (sec)** | If a persistent switch is installed for the selected axis, enter the time required in seconds for the switch to transition to the cooled (persistent) state. |
| **Switch Heating Time (sec)** | If a persistent switch is installed for the selected axis, enter the time required in seconds for the switch to transition to the heated (non-persistent, charging) state. |

## 3.2 Options

The Options dialog (Preferences in the macOS version) provides control of some application-wide settings. These settings are saved between application executions.




A detailed description of each parameter is provided in the following table:

| | |
|---|---|
| **Enter Persistence checked by default on file imports** | If this item is checked, then the **Enter Persistence?** checkbox is checked for every table row during a vector or polar table CSV-file import. |
| **Settling Time (sec) at field before entering persistence** | Once the magnet field reaches the target value and the system moves to the HOLDING state, this value specified how long the application should wait before automatically entering persistence mode if selected for that target. Valid values are 20 to 9999 seconds, inclusive. This ensures all ramping transients have fully decayed before entering persistence. |
| **Magnet-DAQ Executable Location** | Enter or use the file chooser (via the ... icon) to specify the location of the Magnet-DAQ executable in your filesystem. *Each activated axis* will have an associated instance of Magnet-DAQ launched on Connect. |
| **Launch instances of Magnet-DAQ in "minimized" state** | If checked, the instances of Magnet-DAQ will be launched in the hidden state to the taskbar of the operating system (they can be unhidden by selecting them in the taskbar). |

| | |
|---|---|
| | Otherwise, the last window state for each instance of Magnet-DAQ will be restored on Connect. This can be useful for restoring a window arrangement on-screen that shows the on-going status of each axis.<br><br>**NOTE:** Not recommended on the macOS operating system. |
| **Auto Stability Mode Override** | If checked, the AUTO Stability Mode is disabled when connecting to a magnet system. This means that the Stability Setting for each Model 430 can be *manually set* to a custom value which will be *unchanged* when connected to this application. **This is not recommended unless explicitly directed to do so by an AMI Technical Representative.** |

# Advanced Features

# 4 Advanced Features

## 4.1 Vector Table File Format

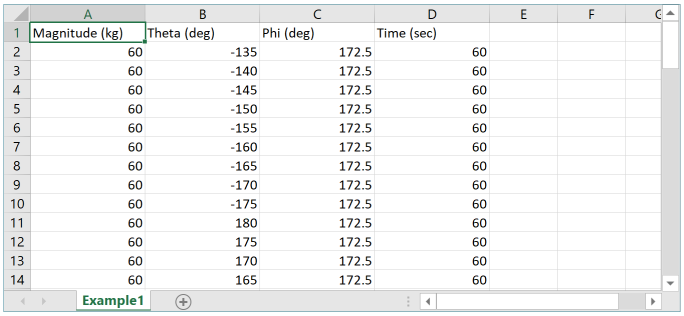
The vector table format supports a basic format and advanced options. The format is Comma-Separated Values (CSV) that is compatible with Excel (or other spreadsheet applications) as well as a simple text editor such as Notepad++.

### Basic Format

The basic format is a file with a single-line, four column header and then subsequent rows of vector values in spherical coordinates and optional dwell time for each in seconds. The format is illustrated below with a spreadsheet view on the left 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.

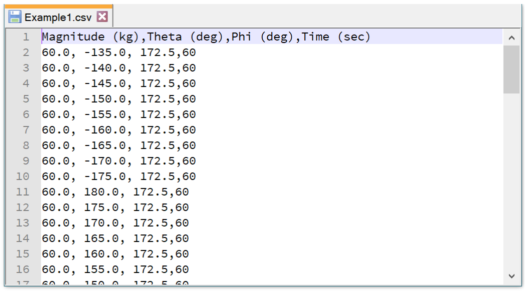
**Example 1:** The header text should be as shown below with the exception that if you wish to use tesla units for the field magnitude, change (kG) to (T). Specifying the units allows the vector table load function to properly identify (and convert if necessary) the field values.

> **NOTE:** By default, the spherical coordinate labeling convention is assumed to be mathematical in the absence of an explicit convention callout in the data file. See Example 3 below.



| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Magnitude (kg) | Theta (deg) | Phi (deg) | Time (sec) | | |
| 2 | 60 | -135 | 172.5 | 60 | | |
| 3 | 60 | -140 | 172.5 | 60 | | |
| 4 | 60 | -145 | 172.5 | 60 | | |
| 5 | 60 | -150 | 172.5 | 60 | | |
| 6 | 60 | -155 | 172.5 | 60 | | |
| 7 | 60 | -160 | 172.5 | 60 | | |
| 8 | 60 | -165 | 172.5 | 60 | | |
| 9 | 60 | -170 | 172.5 | 60 | | |
| 10 | 60 | -175 | 172.5 | 60 | | |
| 11 | 60 | 180 | 172.5 | 60 | | |
| 12 | 60 | 175 | 172.5 | 60 | | |
| 13 | 60 | 170 | 172.5 | 60 | | |
| 14 | 60 | 165 | 172.5 | 60 | | |







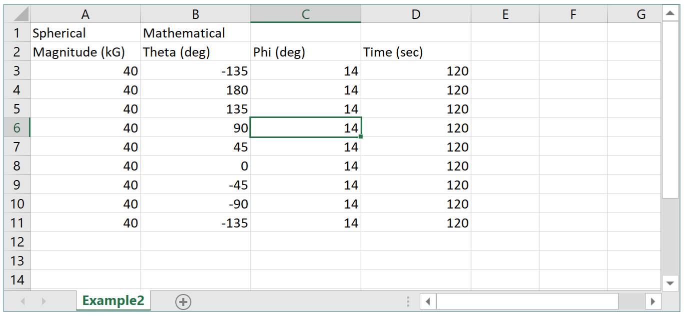
```
Magnitude (kg),Theta (deg),Phi (deg),Time (sec)
60.0, -135.0, 172.5,60
60.0, -140.0, 172.5,60
60.0, -145.0, 172.5,60
60.0, -150.0, 172.5,60
60.0, -155.0, 172.5,60
60.0, -160.0, 172.5,60
60.0, -165.0, 172.5,60
60.0, -170.0, 172.5,60
60.0, -175.0, 172.5,60
60.0, 180.0, 172.5,60
60.0, 175.0, 172.5,60
60.0, 170.0, 172.5,60
60.0, 165.0, 172.5,60
60.0, 160.0, 172.5,60
60.0, 155.0, 172.5,60
```

### Advanced Options

The advanced options supports specifying either Cartesian coordinates, or spherical coordinates in either the mathematical or ISO labeling convention.

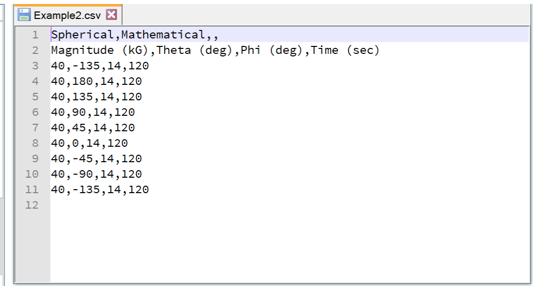
**Example 2:** Explicit specification of spherical coordinates in mathematical convention

Note that the first line contains "Spherical,Mathematical" which explicitly defines the type of coordinates and convention.



| | A | B | C | D |
|---|---|---|---|---|
| 1 | Spherical | Mathematical | | |
| 2 | Magnitude (kG) | Theta (deg) | Phi (deg) | Time (sec) |
| 3 | 40 | -135 | 14 | 120 |
| 4 | 40 | 180 | 14 | 120 |
| 5 | 40 | 135 | 14 | 120 |
| 6 | 40 | 90 | 14 | 120 |
| 7 | 40 | 45 | 14 | 120 |
| 8 | 40 | 0 | 14 | 120 |
| 9 | 40 | -45 | 14 | 120 |
| 10 | 40 | -90 | 14 | 120 |
| 11 | 40 | -135 | 14 | 120 |

Example2



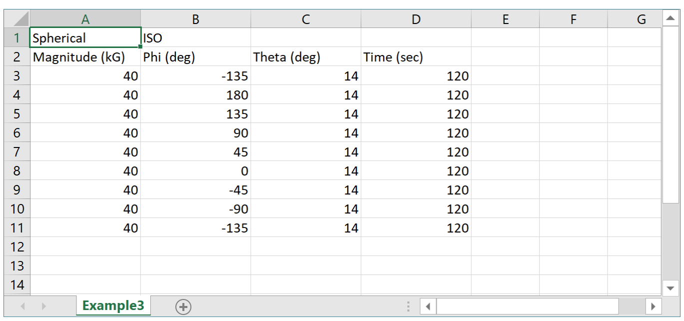
Example2.csv

```
Spherical,Mathematical,,
Magnitude (kG),Theta (deg),Phi (deg),Time (sec)
40,-135,14,120
40,180,14,120
40,135,14,120
40,90,14,120
40,45,14,120
40,0,14,120
40,-45,14,120
40,-90,14,120
40,-135,14,120
```

---

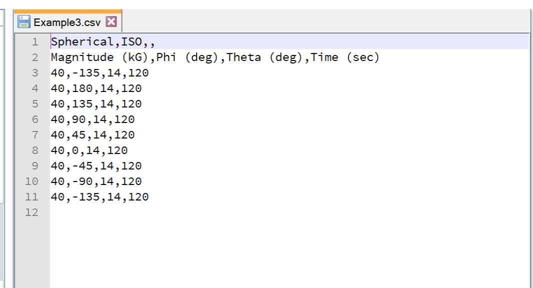
**Example 3:** Explicit specification of spherical coordinates in ISO convention

Note that the first line contains "Spherical,ISO" which explicitly defines the type of coordinates and convention.



| | A | B | C | D |
|---|---|---|---|---|
| 1 | Spherical | ISO | | |
| 2 | Magnitude (kG) | Phi (deg) | Theta (deg) | Time (sec) |
| 3 | 40 | -135 | 14 | 120 |
| 4 | 40 | 180 | 14 | 120 |
| 5 | 40 | 135 | 14 | 120 |
| 6 | 40 | 90 | 14 | 120 |
| 7 | 40 | 45 | 14 | 120 |
| 8 | 40 | 0 | 14 | 120 |
| 9 | 40 | -45 | 14 | 120 |
| 10 | 40 | -90 | 14 | 120 |
| 11 | 40 | -135 | 14 | 120 |

Example3



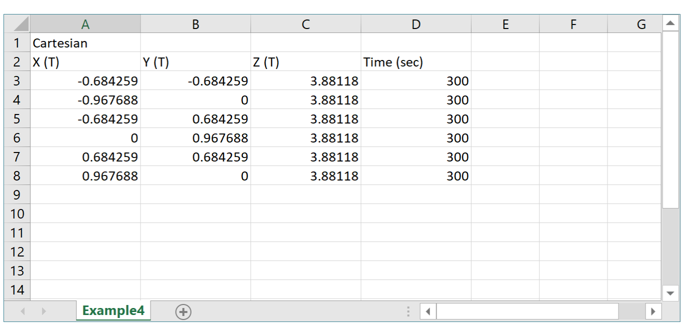
Example3.csv

```
Spherical,ISO,,
Magnitude (kG),Phi (deg),Theta (deg),Time (sec)
40,-135,14,120
40,180,14,120
40,135,14,120
40,90,14,120
40,45,14,120
40,0,14,120
40,-45,14,120
40,-90,14,120
40,-135,14,120
```

---

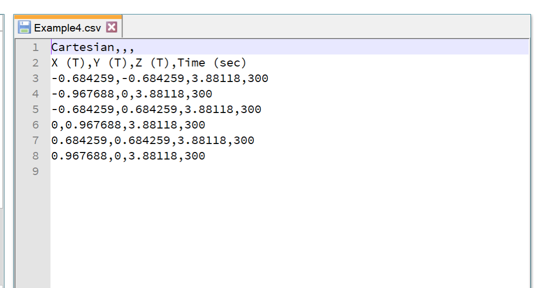
**Example 4:** Cartesian coordinates

Note that the first line contains "Cartesian" which explicitly defines the type of coordinates. Note that field units are in tesla.



| | A | B | C | D |
|---|---|---|---|---|
| 1 | Cartesian | | | |
| 2 | X (T) | Y (T) | Z (T) | Time (sec) |
| 3 | -0.684259 | -0.684259 | 3.88118 | 300 |
| 4 | -0.967688 | 0 | 3.88118 | 300 |
| 5 | -0.684259 | 0.684259 | 3.88118 | 300 |
| 6 | 0 | 0.967688 | 3.88118 | 300 |
| 7 | 0.684259 | 0.684259 | 3.88118 | 300 |
| 8 | 0.967688 | 0 | 3.88118 | 300 |

Example4



Example4.csv

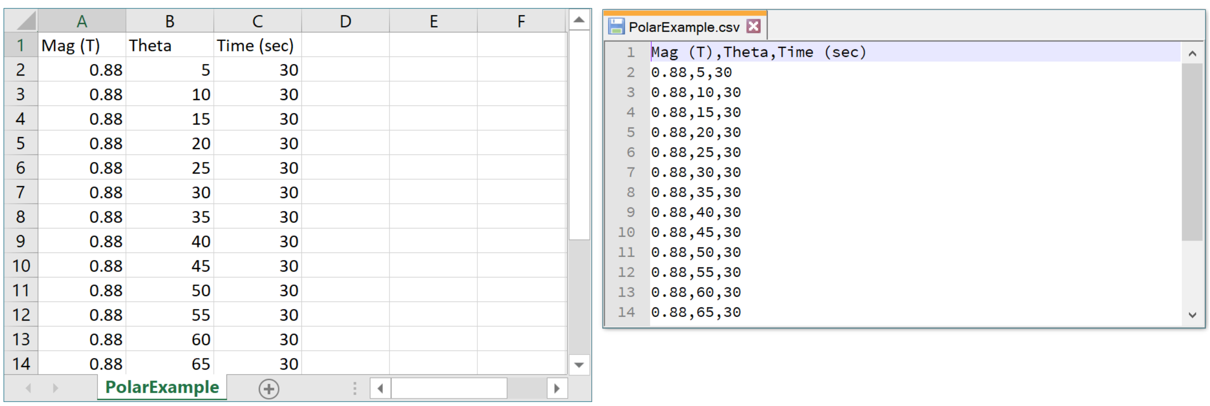
```
Cartesian,,,
X (T),Y (T),Z (T),Time (sec)
-0.684259,-0.684259,3.88118,300
-0.967688,0,3.88118,300
-0.684259,0.684259,3.88118,300
0,0.967688,3.88118,300
0.684259,0.684259,3.88118,300
0.967688,0,3.88118,300
```

## 4.2 Polar Table File Format

The polar table file 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, three column header and then subsequent rows of vector values in polar coordinates and optional dwell time for each in seconds. The format is illustrated below with a spreadsheet view on the left 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:** The header text should be as shown below with the exception that if you wish to use kilogauss units for the field magnitude, change (T) to (kG). Specifying the units allows the polar table load function to properly identify (and convert if necessary) the field values.



| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Mag (T) | Theta | Time (sec) | | | |
| 2 | 0.88 | 5 | 30 | | | |
| 3 | 0.88 | 10 | 30 | | | |
| 4 | 0.88 | 15 | 30 | | | |
| 5 | 0.88 | 20 | 30 | | | |
| 6 | 0.88 | 25 | 30 | | | |
| 7 | 0.88 | 30 | 30 | | | |
| 8 | 0.88 | 35 | 30 | | | |
| 9 | 0.88 | 40 | 30 | | | |
| 10 | 0.88 | 45 | 30 | | | |
| 11 | 0.88 | 50 | 30 | | | |
| 12 | 0.88 | 55 | 30 | | | |
| 13 | 0.88 | 60 | 30 | | | |
| 14 | 0.88 | 65 | 30 | | | |



```
Mag (T),Theta,Time (sec)
0.88,5,30
0.88,10,30
0.88,15,30
0.88,20,30
0.88,25,30
0.88,30,30
0.88,35,30
0.88,40,30
0.88,45,30
0.88,50,30
0.88,55,30
0.88,60,30
0.88,65,30
```

## 4.3 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 |
|---|---|
| -p | Start the stdin/stdout parser function (for Python or QProcess use). |
| --simulate *address* | Places the application in simulation mode with connections to *address* (for AMI-proprietary |

| | |
|---|---|
| | simulation/testing use only). |

**Example:**

**"C:\Program Files\American Magnetics, Inc\Multi-Axis Operation\Multi-Axis-Operation.exe" -p"**

This starts the application and activates the stdin/stout parser.

## 4.4 Remote Scripting Support

The Multi-Axis Operation 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. In fact, any method that can utilize *stdin* and *stdout* pipes can be used to communicate, including LabVIEW.

> **NOTE:** The *Multi-Axis Operation* application itself uses the parser function of Magnet-DAQ application and QProcess objects to operate up to three separate Model 430 units in tandem to achieve 3D vector control.

**This application scripting interface allows multi-axis control of AMI power supply systems *without* having to deal with TCP/IP sockets or serial port programming.** 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 up to 10 significant digits to the right of the decimal (trailing insignificant zeros are omitted) and may be in either decimal or exponential format, whichever is shorter. All returned responses 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. Comma-delimited parameters do not *require* space(s) after the comma, but spaces are acceptable. Commands do not return a value.

> **NOTE:** 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

**`SYSTem:CONNect`**

Initiates a connection to all system devices for the enabled axes as defined by the IP addresses in the Magnet Parameters/Limits panel. Upon successful connection, the system **`STATE?`** will change to "PAUSED". If the connection is not successful, an error will be generated and the system state will remain as "DISCONNECTED".

**`SYSTem:DISConnect`**

Disconnects from all system devices and leaves the application in the DISCONNECTED state.

**`*CLS`**

Clears the error queue.

**`EXIT`**

Disconnects from all system devices and gracefully exits the application.

### SETTINGS COMMANDS

> **NOTE:** Parser commands to configure or query the numerous Magnet Parameter/Limits are not provided. The recommended process is to enter the magnet parameters using the application GUI and then use the *File | Save Settings...* menu item to save the information to a .SAV file (it is suggested to employ a file naming scheme containing the magnet and sample IDs, for example). Then subsequent remote control sessions can use the **`LOAD:SETtings`** command as documented below to restore *all the settings* for the target magnet system, including the sample alignment values and all table contents.

**`LOAD:SETtings`** ***<filename>***

Loads a .SAV settings file from the file system location specified in *filename*. This restores *all* Magnet Parameters/Limits, the Sample Alignment settings, and all table contents for both the Vector and Polar tables.

**`SAVE:SETtings <filename>`**

Creates a .SAV settings file at the file system location specified in *filename*. This saves *all* Magnet Parameters/Limits, the Sample Alignment settings, and all table contents for both the Vector and Polar tables.

**`CONFigure:UNITS {0|1}`**

Sets the preferred field units. Sending "0" selects kilogauss. A "1" selects tesla. The selected field units are immediately applied to both the Multi-Axis Operation display and all the related parser commands and queries that use or return a field value. *The units may only be changed in the disconnected state.*

---

## SAMPLE ALIGNMENT COMMANDS

**`CONFigure:ALIGN1 <magnitude (kG, T)>, <azimuth (deg)>, <inclination (deg)>`**

Sets the Sample Alignment Vector #1 in spherical coordinates relative to the physical magnet axes. The magnitude is specified in the present field units (see **`CONFigure:UNITS`** above), the azimuth angle projected in the X-Y plane as measured from positive X is specified in degrees, and the inclination angle from the positive Z axis is specified in degrees. The plane of rotation for the polar coordinates is immediately updated and the application initiates ramping to the vector.

**`CONFigure:ALIGN2 <magnitude (kG, T)>, <azimuth (deg)>, <inclination (deg)>`**

Sets the Sample Alignment Vector #2 in spherical coordinates relative to the physical magnet axes. The magnitude is specified in the present field units (see **`CONFigure:UNITS`** above), the azimuth angle projected in the X-Y plane as measured from positive X is specified in degrees, and the inclination angle from the positive Z axis is specified in degrees. The plane of rotation for the polar coordinates is immediately updated and the application initiates ramping to the vector.

---

## TARGET VECTOR COMMANDS

**NOTE:** All field values are checked against the Magnet Parameters/Limits. Any commanded vector that violates the limits of the magnet will be rejected with an error and the command will be ignored.

**CONFigure:TARGet:ALIGN1**

Sets the Target Vector to the Sample Alignment Vector #1 and initiates ramping to the vector. This command can be used to return to the alignment vector for verification without changing its value.

**CONFigure:TARGet:ALIGN2**

Sets the Target Vector to the Sample Alignment Vector #2 and initiates ramping to the vector. This command can be used to return to the alignment vector for verification without changing its value.

**CONFigure:TARGet:VECtor *<magnitude (kG, T)>*, *<azimuth (deg)>*, *<inclination (deg)>*, *<time (sec)>***

Sets the Target Vector in spherical coordinates relative to the physical magnet axes. The magnitude is specified in the present field units (see **CONFigure:UNITS** above), the azimuth angle projected in the X-Y plane as measured from positive X is specified in degrees, and the inclination angle from the positive Z axis is specified in degrees. The last parameter, a dwell "time", is optional. Omission of the time value will generate a zero entry in the table for time (i.e. time is ignored).

This command will add the vector to the Vector Table and system initiates immediate ramping to the vector.

EXAMPLE:

**CONF:TARG:VEC 2.0,10.0,15,60**

With selected field units of tesla, the above command will add the spherical vector of magnitude 2.0 T, azimuth 10 degrees, and inclination of 15 degrees to the Vector Table, with a dwell time of 60 seconds, and initiate immediate ramping to that field vector.

**CONFigure:TARGet:VECtor:CARTesian *<xvalue (kG, T)>*, *<yvalue (kG, T)>*, *<zvalue (kG, T)>*, *<time (sec)>***

Sets the Target Vector in Cartesian coordinates relative to the physical magnet axes. The magnitude or each x, y, and z value is specified in the present field units (see **CONFigure:UNITS** above). The last parameter, a dwell "time", is optional. Omission of the time value will generate a zero entry in the table for time (i.e. time is ignored).

This command will add the vector to the Vector Table and system initiates immediate ramping to the vector.

EXAMPLE:

**CONF:TARG:VEC:CART 0.5,0.5,5.0**

With selected field units of tesla, the above command will add the Cartesian vector of (x, y, z) = (0.5 T, 0.5 T, 5.0 T) to the Vector Table, with a dwell time of zero, and initiate immediate ramping to that field vector.

**CONFigure:TARGet:VECtor:TABle *<row #>***

Sets the Target Vector to the integer valued row # which corresponds to a valid entry in the Vector Table and initiates immediate ramping to that field vector. This command is useful if a known Vector Table has been previously loaded via the **LOAD:SETtings** command. Specifying a non-existent row value for the Vector Table will generate an error and the command will be ignored.

**CONFigure:TARGet:POLar *<magnitude (kG, T)>, <angle (deg)>, <time (sec)>***

Sets the Target Vector in polar coordinates in the sample alignment plane. The magnitude is specified in the present field units (see **CONFigure:UNITS** above) and the polar angle relative to the ALIGN1 vector is specified in degrees. The last parameter, a dwell "time", is optional. Omission of the time value will generate a zero entry in the table for time (i.e. time is ignored).

This command will add the vector to the Polar Table and system initiates immediate ramping to the vector.

EXAMPLE:

**CONF:TARG:POL 8.8,56.4**

With selected field units of kilogauss, the above command will add the polar vector of magnitude 8.8 kG and polar angle of 56.4 degrees and initiate immediate ramping to that field vector. The vctor is added to the Polar Table with a dwell time of zero.

**CONFigure:TARGet:POLar:TABle *<row #>***

Sets the Target Vector to the integer valued row # which corresponds to a valid entry in the Polar Table and initiates immediate ramping to that field vector. This command is useful if a known Polar Table has been previously loaded via the **LOAD:SETtings** command. Specifying a non-existent row value for the Polar Table will generate an error and the command will be ignored.

### CONTROL COMMANDS

**PAUSE**

Pauses all connected devices at the present operating field.

**RAMP**

Initiates ramping of all enabled and connected magnet axes to the Target Vector. During ramping, the **STATE?** query will return "RAMPING" status. Upon arrival at the Target Vector the system state will change to "HOLDING".

**ZERO**

Initiates return of the magnet to zero field at the maximum ramp rate.

**PERSistent {0|1}**

Controls the state of the persistent switch. If a value of '0' is sent, the switch is heated (persistent mode is exited). If a value of '1' is sent, the switch is cooled and the system enters persistent mode. You must exit persistent mode to set a new Target Vector, otherwise an error is generated.

## Queries

### SYSTEM QUERIES

***IDN?**

Returns the identification string of the Multi-Axis Operation application. The identification string contains the app name and revision level of app. Example output:
**MultiAxisOperation,1.00**

**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.

`UNITS?`

Returns the present field units active for the application. A return value of "0" indicates units of kilogauss. A return value of "1" indicates tesla.

`PERSistent?`

Returns the state of the persistent switch. A return value of '0' indicates the switch is heated (persistent mode is exited). A return value of '1' is indicates the switch is cooled and the system is in persistent mode.

`STATE?`

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

| Return Values for `STATE?` Query | |
|---|---|
| Return Value | Meaning |
| **0** | DISCONNECTED |
| **1** | RAMPING to target field |
| **2** | HOLDING at the target field |
| **3** | PAUSED |
| **4** | ZEROING FIELD (in progress) |
| **5** | At ZERO field |
| **6** | Quench detected |
| **7** | Heating persistent switch(es) |
| **8** | Cooling persistent switch(es) |

## SAMPLE ALIGNMENT QUERIES

`ALIGN1?`

Returns the Sample Alignment Vector #1 value in spherical coordinates relative to the physical magnet axes in the selected field units and degrees.

`ALIGN1:CARTesian?`

Returns the Sample Alignment Vector #1 value in Cartesian coordinates relative to the physical magnet axes in the selected field units.

`ALIGN2?`

Returns the Sample Alignment Vector #2 value in spherical coordinates relative to the physical magnet axes in the selected field units and degrees.

**ALIGN2:CARTesian?**

Returns the Sample Alignment Vector #2 value in Cartesian coordinates relative to the physical magnet axes in the selected field units.

**PLANE?**

Returns the 3D implicit plane equation coefficients (A,B,C), where A x + B y + C z = 0, which is also the normalized cross product result of the two sample alignment vectors.

### TARGET VECTOR QUERIES

**TARGet?**

Returns the Target Vector (i.e. target) value in comma-delimited spherical coordinates relative to the physical magnet axes in the selected field units and degrees.

**TARGet:CARTesian?**

Returns the Target Vector (i.e. target) value in comma-delimited Cartesian coordinates relative to the physical magnet axes in the selected field units.

**TARGet:TIME?**

Returns an *estimate* of the remaining time in seconds until the Target Vector is realized. The actual time required typically may be 5 to 30 seconds greater than the estimate, depending on the specific system configuration.

### PRESENT MAGNET FIELD QUERIES

**FIELD?**

Returns the present magnet field in comma-delimited spherical coordinates relative to the physical magnet axes in the selected field units and degrees.

**FIELD:CARTesian?**

Returns the present magnet field value in comma-delimited Cartesian coordinates relative to the physical magnet axes in the selected field units.

## Error Codes

The following are error codes returned by the **SYSTem:ERRor?** query. The queue is last-in, first-out (LIFO).

**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.

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

A parameter is in a non-numerical format.

**`-152,"Magnitude exceeds limit"`**

The field vector magnitude exceeds the limit for the magnet.

**`-153,"Negative magnitude"`**

Negative magnitude value is not allowed.

**`-154,"Inclination out of range"`**

The inclination angle is out of range. The inclination angle is specified in degrees from the positive z-coil axis, and must be greater than or equal to zero and less than or equal to 180.

**`-155,"Field exceeds x-coil limit"`**

The x-component of the field vector exceeds the x-coil limit.

**`-156,"Field requires x-coil"`**

The field vector requires an x-coil component, but the x-coil is not active.

**`-157,"Field exceeds y-coil limit"`**

The y-component of the field vector exceeds the y-coil limit.

`-158,"Field requires y-coil"`

The field vector requires a y-coil component, but the y-coil is not active.

`-159,"Field exceeds z-coil limit"`

The z-component of the field vector exceeds the z-coil limit.

`-160,"Field requires z-coil"`

The field vector requires a z-coil component, but the z-coil is not active.

---

### QUERY ERRORS

`-201,"Unrecognized query"`

The query is unrecognized.

---

### EXECUTION ERRORS

`-301,"Not connected"`

The system is in the DISCONNECTED state.

`-302,"Switch in transition"`

The requested action is not allowed while a persistent switch is in transition.

`-303,"Quench condition"`

The action is not allowed while the system is in the QUENCHED state. Clear the quench condition to continue.

`-304,"No units change while connected"`

Changing field units is not allowed once the system is in the CONNECTED state.

`-305,"Cannot enter persistence"`

Cannot enter persistent mode unless the system is in a PAUSED or HOLDING state.

`-306,"System is persistent"`

The action is not allowed while the system is in persistent mode.

`-307,"No switch installed"`

The requested action is invalid without an installed persistent switch.

```
-308,"Cannot LOAD while connected"
```

The LOAD SETTINGS action is not allowed once the system is in the CONNECTED state.

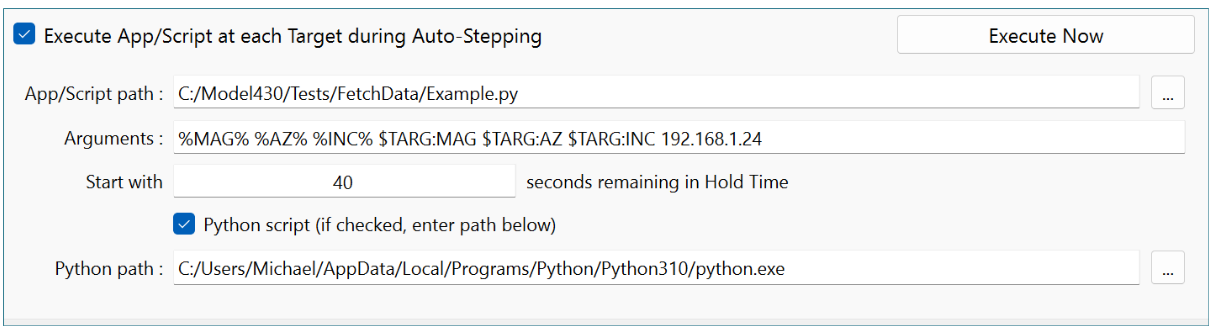
## 4.5 Python Examples

### 4.5.1 Example Python Script

The following is the Example.py Python script as referenced in the Vector and Polar tables descriptions. Such a script shows how the operator can leverage this feature to execute repetitive DAQ tasks at each vector. This example supports any digital multimeter that is HP/Agilent 34401A compatible, which includes most recent Keysight models.

This simple example script accepts an optional last 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.

It uses the Arguments field to pass the present field state and target field to the script for output to the data table.




*Execution using special variables as Arguments*

Also, note in the script that if you provide a non-zero return code on sys.exit(), then the last print() text is displayed in the Status Bar area of the application.

```
# 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
```

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

# get args with field magnitude, azimuth, and inclination
magnitude = (str(sys.argv[1]));
azimuth = (str(sys.argv[2]));
inclination = (str(sys.argv[3]));
targmag = (str(sys.argv[4]));
targaz = (str(sys.argv[5]));
targinc = (str(sys.argv[6]));

# check to see if eight arguments provided, should be last arg with ip a
if len(sys.argv) > 7:
    addrDVM = str(sys.argv[7])
else:
    addrDVM = "192.168.1.17"    # default if not provided

# connect to DVM
try:
    DVM.connect((addrDVM, 5025))
except:
   # this message will display in the status bar of MAO app if the exit
   print("Could not connect to DVM @ " + addrDVM)
   sys.exit(1)

#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(magnitude + "," + azimuth + "," + inclination + "," + tar
datafile.write(data.decode("utf-8").strip('\n'))
datafile.write('\n');

# close datafile
datafile.close()
```

```
# close socket
DVM.close()

# exit with success code
sys.exit(0)
```

### 4.5.2 Automated App Control Example 1

The following is a Python 3.6 example for controlling the Multi-Axis Operation application with an interactive external script:

```
Python 3.6 (64-bit) interactive window [PTVS 15.7.18156.1-15.0]
Type $help for a list of commands.
>>> import subprocess
>>> p = subprocess.Popen('C:\Program Files\American Magnetics,
Inc\Multi-Axis Operation\Multi-Axis-Operation -p',
stdin=subprocess.PIPE, stdout=subprocess.PIPE, bufsize = 1)
>>> p.stdin.write(b'LOAD:SET C:/Model430/Tests/14786-switch.sav\n')
44
>>> p.stdin.flush()
>>> p.stdin.write(b'SYST:CONN\n')
10
>>> p.stdin.flush()
>>> p.stdin.write(b'ZERO\n')
5
>>> p.stdin.flush()
>>> p.stdin.write(b'conf:targ:vec:table 1\n')
22
>>> p.stdin.flush()
>>> p.stdin.write(b'targ?\n')
6
>>> p.stdin.flush()
>>> print(p.stdout.readline())
b'4,-135,14\r\n'
>>> p.stdin.write(b'targ:cartesian?\n')
16
>>> p.stdin.flush()
>>> print(p.stdout.readline())
b'-0.6842584516,-0.6842584516,3.881182905\r\n'
>>>
```

The texts in bold above are command entries. The above code performs the following steps:

1) Launches the application with the parser enabled (-p argument) which enables communication via pipes

2) Loads the complete application settings from the file **C:/Model430/Tests/14786-switch.sav**
3) Initiates a connection to the power supply system components
4) Commands the system to ZERO field
5) Chooses table row 1 as the Target Vector
6) Readback of the Target Vector in spherical coordinates
7) Readback of the Target Vector in Cartesian coordindates

### 4.5.3 Automated App Control Example 2

The following is a Python 3.10 example for controlling the Multi-Axis Operation application with an external script.

The script launches the Multi-Axis Operation app, loads a settings file, and then loops through the first five Vector Table field points. It waits 20 seconds once each field point is reached before proceeding to the next.

```
import subprocess
import time
import sys

# launch the Multi-Axis Operation (MAO) app
px = subprocess.Popen('C:/Program Files/American Magnetics, Inc/Multi-Ax

# sleep to allow app to start and parser to initialize
time.sleep(20)

# query for MAO version info
px.stdin.write(b'*IDN?\n')

px.stdin.flush()
print(px.stdout.readline().strip())

# load the settings file
px.stdin.write(b'LOAD:SET C:/Model430/Customers/Magnet-14938/Magnet-1493
px.stdin.flush()

# proceed to connect to the system
px.stdin.write(b'SYSTEM:CONNECT\n')
px.stdin.flush()

print("Waiting for CONNECT", end='')

stateVal = 0
stateStr = ""

# checking for a connected state
```

```
while not stateVal:
  time.sleep(1)
  px.stdin.write(b'STATE?\n')

  px.stdin.flush()
  stateStr = px.stdout.readline()
  stateVal = int(stateStr.strip())

  if not stateVal:
    print('.', end='')
  else:
    print("STATE = " + str(stateVal))

row_index = 1

# loop through the first five field points in the Vector Table
while row_index <= 5:
    # go to the vector at row_index in the Vector Table (based on a star
    # you need foreknowledge of the field points in the stored/loaded Ve
    # e.g. the Vector Table could be a common field point list for a giv
    # otherwise, use CONFigure:TARGet:VECtor (see MAO Help)

    str_index = str(row_index)
    print("Go to field at row = " + str_index + " in Vector Table")
    px.stdin.write(b'CONFIGURE:TARGET:VECTOR:TABLE ' + str_index.encode(
    px.stdin.flush()

    # short delay to allow new target to register, then read STATE
    time.sleep(2)
    px.stdin.write(b'STATE?\n')
    px.stdin.flush()
    stateStr = px.stdout.readline()
    stateVal = int(stateStr.strip())

    # check the system state and look for HOLDING (2)
    while stateVal != 2:
      time.sleep(10)
      px.stdin.write(b'STATE?\n')
      px.stdin.flush()
      stateStr = px.stdout.readline()
      stateVal = int(stateStr.strip())

      if stateVal != 2:
        print('.', end='')
      else:
        print("STATE = " + str(stateVal))

    # have reached field!
    # put your custom DAQ here instead of a time delay
    if row_index == 5:
        print("Have reached field! Hold for 20 seconds and then exit.")
    else:
        print("Have reached field! Hold for 20 seconds and then continue
```

```
    time.sleep(20)
    row_index += 1;

# gracefully exit
px.stdin.write(b'EXIT\n')

# return success code
sys.exit(0)
```

Example output from the above script:

```
b'MultiAxisOperation,1.03'
Waiting for CONNECT.............STATE = 3
Go to field at row = 1 in Vector Table
............................................................................................
...................STATE = 2
Have reached field! Hold for 20 seconds and then continue.
Go to field at row = 2 in Vector Table
............................................................................................
..................STATE = 2
Have reached field! Hold for 20 seconds and then continue.
Go to field at row = 3 in Vector Table
............................................................................................
..................STATE = 2
Have reached field! Hold for 20 seconds and then continue.
Go to field at row = 4 in Vector Table
............................................................................................
..................STATE = 2
Have reached field! Hold for 20 seconds and then continue.
Go to field at row = 5 in Vector Table
............................................................................................
..................STATE = 2
Have reached field! Hold for 20 seconds and then exit.
```

# Support

# 5 Support

## 5.1 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/multi-axis-operation

Ready-to-use binaries are provided on the Downloads page for 64-bit versions of Windows 7/8/10/11, Linux (Ubuntu 18.04 or later recommended), and Apple macOS 10.10 (Yosemite) 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

# 6 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 -



# - G -



# - I -



# - L -



# - M -



# - O -



# - P -



# - Q -



# - R -



# - S -



# - T -



# - V -



# - Z -