

Software Manual

# MeasureLINK™ MCS Software

Lake Shore Cryotronics, Inc.  
575 McCorkle Blvd.  
Westerville, Ohio  
43082-8888 USA

[sales@lakeshore.com](mailto:sales@lakeshore.com)  
[support@lakeshore.com](mailto:support@lakeshore.com)

[www.lakeshore.com](http://www.lakeshore.com)

Fax: (614) 891-1392  
Telephone: (614) 891-2243



Methods and apparatus disclosed and described herein have been developed solely on company funds of Lake Shore Cryotronics, Inc. No government or other contractual support or relationship whatsoever has existed which in any way affects or mitigates proprietary rights of Lake Shore Cryotronics, Inc. in these developments. Methods and apparatus disclosed herein may be subject to U.S. Patents existing or applied for.

Lake Shore Cryotronics, Inc. reserves the right to add, improve, modify, or withdraw functions, design modifications, or products at any time without notice. Lake Shore shall not be liable for errors contained herein or for incidental or consequential damages in connection with furnishing, performance, or use of this material.

Rev. 2.5

4 September 2024

## **CERTIFICATION**

Lake Shore certifies that this product has been inspected and tested in accordance with its published specifications and that this product met its published specifications at the time of shipment. The accuracy and calibration of this product at the time of shipment are traceable to the United States National Institute of Standards and Technology (NIST); formerly known as the National Bureau of Standards (NBS).

## **TRADEMARK ACKNOWLEDGMENT**

Many manufacturers and sellers claim designations used to distinguish their products as trademarks. Where those designations appear in this manual and Lake Shore was aware of a trademark claim, they appear with initial capital letters and the ™ or ® symbol.

Microsoft Windows®, Visual Basic, and Excel® are registered trademarks of Microsoft Corporation in the United States and other countries.

Nycoil® is a registered trademark of the Nycoil Company Corporation, New York.

Apiezon® is a registered trademark of M&I Materials, Ltd.

Kimwipes® is a registered trademark of Kimberly-Clark Worldwide.

Kapton® is a trademark of DuPont De Nemours.

Dust-Off™ is a registered trademark of Falcon Safety Products, Inc.

WinWrap® Basic is a registered trademark of Polar Engineering, Inc.

MultiVu™ is a trademark of Quantum Design.

Copyright 2019 – 2024 Lake Shore Cryotronics, Inc. All rights reserved. No portion of this manual may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the express written permission of Lake Shore.



# Table of Contents

## Chapter 1: Software Operation

1.1	General	1
1.2	MeasureLINK™ Concepts	1
1.2.1	Getting Started with MeasureLINK™	1
1.2.1.1	Update Application Packs	1
1.2.1.2	Configure Instruments	1
1.2.1.3	Build a Sequence	1
1.2.1.4	Modify a Script	1
1.2.1.5	Export Data	1
1.2.2	File Types	2
1.2.2.1	Measurement Sequence Files (*.mlseq)	2
1.2.2.2	Script Files (*.mlscr)	2
1.2.3	MeasureLINK™ Instrument Communication	2
1.2.3.1	System Configuration	3
1.2.3.2	Drivers	3
1.2.3.3	Communications Interfaces	3
1.3	Understanding the User Interface	5
1.3.1	Ribbon	5
1.3.2	Navigation Pane	5
1.3.3	Workspace	5
1.3.4	Monitor Window	6
1.3.5	Other Indicators	7
1.3.6	Backstage View	7
1.3.6.1	New	7
1.3.6.2	Open...	7
1.3.6.3	Recent	7
1.3.6.4	Help	7
1.3.6.5	Close	8
1.4	Creating New Measurements	9
1.4.1	Creating a New Measurement File	9
1.4.2	Opening an Existing Measurement File	9
1.4.3	Naming and Saving a Measurement File	9
1.5	Measurement Sequences	10
1.5.1	Sequence Measurement Concepts	10
1.5.2	Sequence Measurement Window	11
1.5.3	Sequence Measurement Controls	13
1.5.3.1	Measurement	13
1.5.3.2	Insert Steps	16
1.5.3.3	Clipboard	17
1.5.3.4	Move	17
1.5.3.5	Tools	17
1.6	Types of Sequence Steps	18
1.6.1	Measurements	18
1.6.1.1	Measure Temperature Sequence Step	18
1.6.2	Loops	19
1.6.2.1	Iteration Loop Sequence Step	19
1.6.2.2	Continuous Field Loop Sequence Step	20
1.6.2.3	Discrete Field Loop Sequence Step	21
1.6.2.4	Continuous Temperature Loop Sequence Step	23
1.6.2.5	Discrete Temperature Loop Sequence Step	25
1.6.3	Controls	26
1.6.3.1	Go to Field with Chart Sequence Step	26
1.6.3.2	Go to Temperature with Chart Sequence Step	27

1.6.3.3	Pause Sequence Step .....	28
1.6.3.4	Sequence Group Step .....	29
1.6.3.5	Sequence Join Step .....	30
1.6.3.6	Sequence Finish Step .....	32
1.6.4	Data .....	33
1.6.4.1	Chart Sequence Step .....	33
1.6.4.2	Analysis Sequence Step .....	38
1.7	Scripts .....	41
1.7.1	Script Measurement Concepts .....	41
1.7.2	Script Window Features .....	41
1.7.2.1	Script Selection Tabs .....	42
1.7.2.2	Results Pane .....	42
1.7.2.3	Script Editor Pane .....	42
1.7.2.4	Script Navigation .....	43
1.7.2.5	Debugging Pane .....	43
1.7.2.6	Ribbon .....	44
1.7.3	Referencing Hardware .....	45
1.7.4	User Interface Scripts .....	46
1.7.5	Data Acquisition Scripts .....	46
1.7.6	Debugging a Script .....	47
1.7.6.1	Debug.Print .....	47
1.7.6.2	Breakpoints .....	47
1.7.6.3	Stopping the Script Execution .....	48
1.7.6.4	Stepping through the Code .....	48
1.7.6.5	Using the Debugging Pane Tabs .....	48
1.7.7	Registering Script Files .....	50
1.7.7.1	Registering a Script .....	50
1.7.7.2	Editing a Script Registration .....	51
1.7.7.3	Deleting a Script Registration .....	51
1.8	Home Level .....	52
1.8.1	Getting Started .....	52
1.8.2	Measurement Registration .....	52
1.8.3	System Configuration .....	53
1.8.3.1	Configuration Controls .....	53
1.8.3.2	Device Settings .....	55
1.8.3.3	System Tool Bar Settings .....	56
1.8.3.4	Process View .....	65
1.8.4	User Preferences .....	66
1.9	Utilities .....	68
1.9.1	Application Packs .....	68
1.9.1.1	Application Pack Utility Overview .....	68
1.9.1.2	Installing Application Packs .....	69
1.9.1.3	Removing Application Packs .....	69
1.9.1.4	Repairing Application Packs .....	69
1.9.1.5	Staged Operations on Restart .....	70
1.9.2	File Recovery .....	71
1.9.3	License Management .....	72
1.9.3.1	Current License .....	72
1.9.3.2	Activation .....	73
1.9.4	Chart Recorder .....	74
1.9.4.1	Chart Recorder Overview .....	74
1.9.4.2	Chart Recorder Configuration Panel .....	75
1.9.4.3	Saving and Restoring Chart Recorder Configurations .....	80

**Chapter 2:  
Remote Operation**

2.1	General .....	81
2.2	Selecting and Configuring an Interface .....	81
2.2.1	TCP-IP Interface .....	81
2.2.2	Enabling Remote Operation in MeasureLINK™ .....	81
2.3	SCPI Language .....	81
2.4	Common Commands .....	81
2.5	Measurement Commands .....	82
2.5.1	Fetch Subsystem .....	82
2.5.2	Source Subsystem .....	82



# Chapter 1: Software Operation

## 1.1 General

This guide explains basic operation for the Lake Shore MeasureLINK™-MCS software.

## 1.2 MeasureLINK™ Concepts

MeasureLINK™-MCS software allows users to construct laboratory measurements combining Lake Shore instruments, third party instruments, and other experimental hardware. It provides a general purpose tool for collecting, storing, processing, and presenting research data. Using the software to run a measurement is simple but there are several key concepts to understand to effectively design and build custom measurements. These concepts can be divided up by the function they serve. Each is discussed in detail later in this document.

The MeasureLINK™ interface is organized based on various activities you might need to perform throughout the duration of your measurement:

- **Measurement Sequences level** (section 1.5): fundamental measurements.
- **Script level** (section 1.7): integrated scripting language based on Microsoft® Visual Basic.NET.
- **Home level** (section 1.8): hardware and software configuration.
- **Utilities level** (section 1.9): utilities for completing various tasks.

### 1.2.1 Getting Started with MeasureLINK™

To get started with MeasureLINK, you will need to perform some basic functions. See the sections below for more detail.

#### 1.2.1.1 Update Application Packs

The Application Pack utility allows you to determine and install the components needed for your measurements. For further information, see section 1.9.1.

#### 1.2.1.2 Configure Instruments

Most measurements require interacting with some sort of instrumentation. MeasureLINK uses device drivers to communicate with instruments and other hardware. Devices are easily installed and configured through MeasureLINK. For further information, see section 1.7.3.

#### 1.2.1.3 Build a Sequence

Most measurements run as a sequence, which is an ordered list of steps to complete a measurement. The user simply edits the UI for each sequence step and then starts the sequence. For further information, see section 1.5.

#### 1.2.1.4 Modify a Script

MeasureLINK hosts an integrated scripting language based on Microsoft® Visual Basic.NET. Pre-defined scripts may also be imported into MeasureLINK. This requires the optional scripting development license (ML-SDL), which can be purchased from Lake Shore. MeasureLINK software and the scripting development license need to be reactivated annually. For further information, see section 1.7.

#### 1.2.1.5 Export Data

MeasureLINK data can be exported in several ways. Scripts and measurement sequences can be exported as .csv files (section 1.5.3.5), and system configurations can be exported as JSON files (section 1.8.3.1.5).

## 1.2.2 File Types

MeasureLINK™-MCS software uses the following extensions to differentiate the type of files:

- .mlseq: Measurement sequence files
- .mlscr: Script files

### 1.2.2.1 Measurement Sequence Files (\*.mlseq)

A measurement sequence is an ordered list of steps to perform to complete a measurement. Measurement steps can be predefined operations or custom steps based on a script. Most measurements run as a sequence.

### 1.2.2.2 Script Files (\*.mlscr)

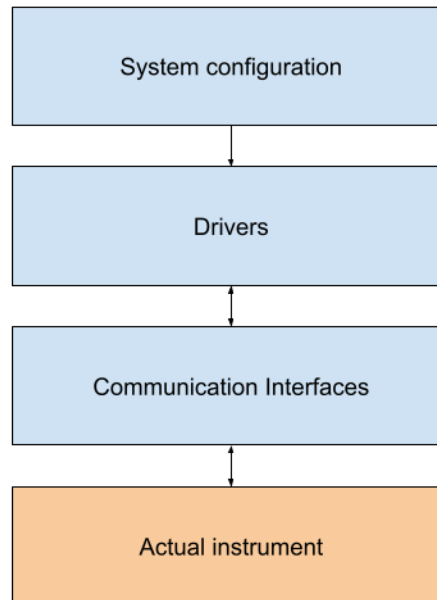
MeasureLINK™ hosts an integrated scripting language based on Microsoft® Visual Basic.NET, which allows the ability to develop custom solutions to collect, process, and present data. External instruments and tools can also be accessed from a script and integrated into your measurements. Once created, scripts can be integrated as sequence steps in a measurement sequence.

MeasureLINK exposes functions to the script language to allow the script to collect data from instruments. The data is then stored, processed, and presented either visually within MeasureLINK or to third-party tools for further processing. For further information on measurement sequences, see section 1.5. For further information on scripts, see section 1.7.

## 1.2.3 MeasureLINK™ Instrument Communication

To take measurements, MeasureLINK™-MCS software must be able to control one or more instruments. MeasureLINK uses three different layers:

- System configuration
- Drivers
- Communications interfaces.



Each is explained in more detail below.



### 1.2.3.1 System Configuration

System configurations contain the MeasureLINK representation of the instruments to be used in one or more measurements. For each required instrument, the system configuration will contain the driver to be used, the preferred interface for communication, and the settings to use on the instrument. It also controls the order in which instruments appear in the Monitor window. Many system configurations can be defined, but only one can be active at a time. System configurations are managed in the System Configuration window. See section 1.8.3 for more information.

### 1.2.3.2 Drivers


Drivers are responsible for communication with an instrument or family of instruments. Drivers take commands from the MeasureLINK software and translate them into calls to the instrument. Data from the instrument is then interpreted by the driver and returned to MeasureLINK. Drivers are responsible for the settings in the system configuration, the instrument programming interface (API), the Monitor window for the instrument, and some utilities. Drivers are typically installed from an application pack (for more information on application packs see section 1.9.1) The list of installed drivers can be viewed on the Help page (in the Backstage view), or from the Add Driver(s) pop-up window in the System Configuration window.

### 1.2.3.3 Communications Interfaces

Each instrument communicates with the host computer through communication interfaces. The interface can be a direct connection like a USB port, or it over a network or through another instrument or software application. The driver for an instrument knows the communications interfaces that instrument supports and how to configure them. Each instrument has a simulator interface, and other interfaces which vary by driver.

#### 1.2.3.3.1 Simulators

Every driver in MeasureLINK provides a simulator interface to test a measurement without using the actual instrument. When the simulator for an instrument is active in the system configuration, the Monitor window for that entry will have a large

S icon  on the title bar and the title text is another color.



**The quality of the simulation varies for each driver, with some being more realistic than others. However, they are usually adequate to test a measurement without access to the actual instrumentation.**

#### 1.2.3.3.2 VISA vs Native communication

Most drivers that support serial or TCP/IP have two communications interfaces for each type of connection: VISA and Native. VISA drivers communicate through the VISA communications standard. VISA (Virtual Instrument Software Architecture) is a standard for communication with instruments via a computer and is controlled by the IVI Foundation. To use VISA, you must have a third-party VISA.NET implementation installed on your computer. NI-VISA from National Instruments, and Keysight IO Libraries are both common VISA.NET implementations. VISA supports robust logging, diagnostics, and less common communication interfaces such as GPIB.

Native interfaces communicate directly through Windows. This removes the need for third-party software to be installed, but also means the diagnostics and logging are not as robust. Native communication is only supported on serial interfaces and TCP/IP.

Performance is very similar between the two communications layers. If a VISA implementation is not installed, then the Native implementation should be used. If VISA is installed, then either option can be chosen.

# 1.3 Understanding the User Interface

This section defines basic terminology and describes the components of the GUI. Gaining an understanding of the GUI organization early will help you learn the system quickly and operate it efficiently.

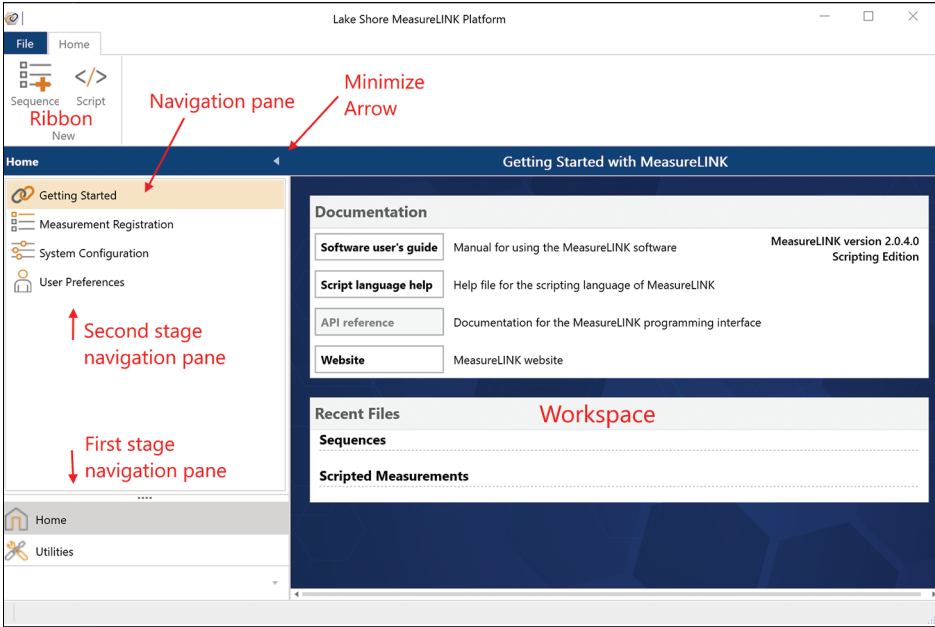


FIGURE 1-1 Interface elements

## 1.3.1 Ribbon

The ribbon is the wide tabbed area that runs across the top of the window. It contains icons for controls, organized into tabs and groups, such as those for starting and stopping measurements and programs, and for editing window setups. The Home tab contains the most common commands. The commands on the ribbon are context specific. This means that the icons and their locations change depending on which workspace is currently displayed. Icons that are unavailable are grayed out. Click any available button to initiate that command.

## 1.3.2 Navigation Pane

The navigation pane occupies most of the left side of the screen (FIGURE 1-1). It provides two stages of navigation. The first stage, always present in the lower corner of the window, allows navigation between the levels.

Second stage navigation, located at the top of the navigation pane, is context-specific and changes depending on the present level that you are viewing. Coloring or highlights in the navigation pane often provide visual feedback of what operation is active. Select any option in the list to go to that function. Minimize the navigation pane to increase the size of your workspace by clicking the arrow at the top of the navigation pane (see FIGURE 1-1).

## 1.3.3 Workspace

The workspace, the large area in the center of the window, is where core work is performed (FIGURE 1-1). Configure measurements and view results in the workspace. Necessary operator interactions are performed here as well. The content and layout of the workspace changes with each level or activity to provide clear and concise feedback. Use the tabs at the bottom of the workspace to select different screens. Use the tabs at the top of the workspace to select different results screens if more than one is generated.

### 1.3.4 Monitor Window

The Monitor window allows selection and control of the devices and instruments in the current system configuration (section 1.8.3). Each instrument in the system configuration has a Monitor display on the Monitor window. This will show the “live” instrument status and possibly allow control of the instrument.

The figure below shows a typical Monitor window. The items displayed in the Monitor window vary based on the instrument that is being used. However, all Monitor displays have some features in common. Each instrument’s Monitor display contains:

- A blue title area containing the type of instrument and its name within MeasureLINK-MCS.
- To the right of the title is a slider switch for controlling connecting and disconnecting from the instrument.
- Below the title box is the grey header region. The header displays the instrument status and the most important data from the instrument.
- On the right side of the header is a small arrow that can be used to expand or collapse the rest of the instrument’s Monitor display.
- The header at the top of the Monitor window contains buttons to expand or collapse all of the Monitor displays. There is also a checkbox called “Keep on top”. When checked, the Monitor window will float above other windows to prevent it from getting covered up.
- Below the header is the System Status area. There is a slider switch here to disable updates for the entire Monitor window, which is useful if the communications overhead of updating the Monitor window is affecting your instrument. The System Status area also shows when a measurement is running. During a measurement, the Monitor window will continue to update if it is not disabled. However, all controls on the Monitor window are read-only while a measurement is in progress. This will keep a change made from the Monitor window from disrupting the measurement.



Simulated instruments appear with an S icon , and appear in a different color.



The options you see may differ, depending on your system configuration.

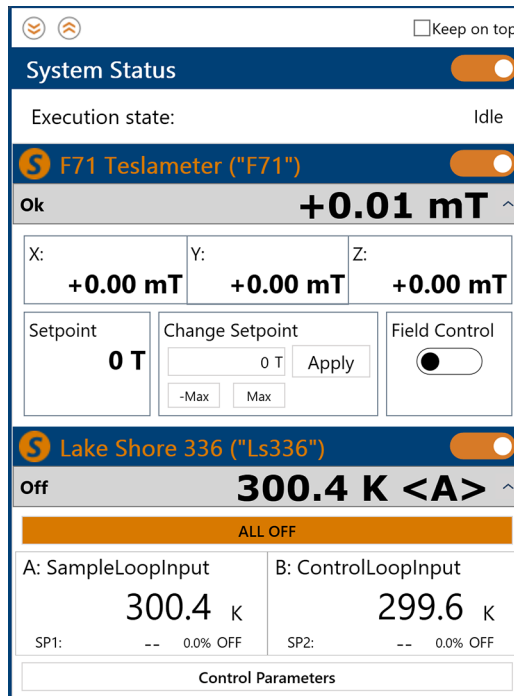


FIGURE 1-2 Monitor window

### 1.3.5 Other Indicators

The status bar is at the lower left corner of the window. It provides the status any time a measurement is running. The status bar does not change based on navigation, so the status of a running measurement can be seen even when the results of an earlier measurement are being viewed in the workspace.

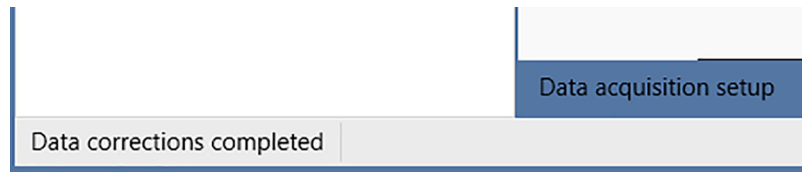


FIGURE 1-3 Status bar

### 1.3.6 Backstage View

The Backstage view is the centralized location for managing files; it contains controls and other features normally associated with application software. Access the Backstage view from the File tab at the top left corner of any window. The File tab is always present in the same location, but some commands may be grayed out when they are unavailable. The same Backstage view is also in each measurement file that is opened. Each of the backstage pages are explained below.

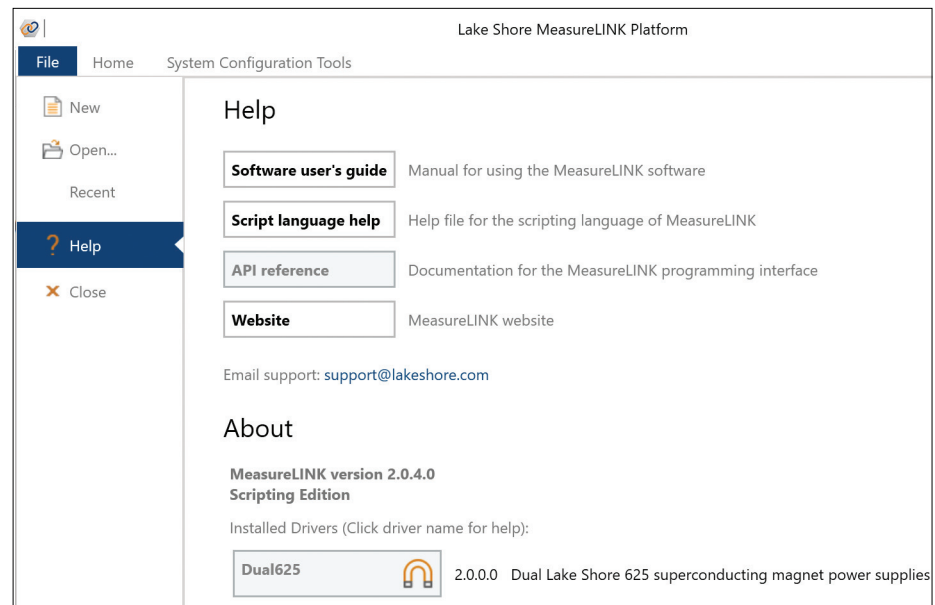


FIGURE 1-4 Typical Backstage view

#### 1.3.6.1 New

This page allows the creation of a new MeasureLINK-MCS™ measurement file (either a script or sequence). This mirrors the main window ribbon buttons for creating scripts or sequences. The new file will open in its own window.

#### 1.3.6.2 Open...

Clicking **Open** displays the standard Windows® Open dialog box to open an existing MeasureLINK™ file.

#### 1.3.6.3 Recent

The Recent page shows the last several files used. Click a file name to re-open the file.

#### 1.3.6.4 Help

The Help page contains documentation and version information. It is split into two sections:

- *Help*: the Help section contains links to this manual, the Script language help file, and the API reference manual for the script API (the programming interface that MeasureLINK™ exposes to script).
- *About*: the About section lists the version information for the MeasureLink™-MCS software, and any installed drivers. If an installed driver has associated documentation, the link is listed in bold type. Click the driver name to see the documentation.

#### 1.3.6.5 Close

The **Close** button will close the current window. If **Close** is clicked in the main window, the entire application will close.

## 1.4 Creating New Measurements

This section explains how to create a measurement file, run a measurement, and view results.

### 1.4.1 Creating a New Measurement File

Create new files by clicking one of the **New** buttons on the ribbon. New files can also be created using the **New** button in the Backstage view.

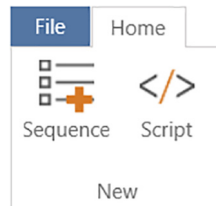


FIGURE 1-5 Ribbon

### 1.4.2 Opening an Existing Measurement File

To open an existing file:

1. Click the **File** tab.
2. In the Backstage view, click **Open**. An Open dialog box appears.
3. Select the file.
4. Click **Open** in the Open dialog box.

The Backstage view also contains a lists the last several files used. Click a file to open it.

### 1.4.3 Naming and Saving a Measurement File

Use **Save** or a **Save As** to save a file:

1. Click **File**, then click **Save** if you are saving it for the first time.
2. Click **Save As** if you are copying a measurement setup file to a new file with a different name.
3. A **Save As** dialog box appears, with a suggested file name. Use this name or change it to another name if you prefer.
4. Choose the desired file location and click **Save**; the default location is: Libraries\Documents\MeasureLINK

In some situations, the MeasureLINK™-MCS software automatically triggers a **Save** or **Save As**. For example, if a measurement is run, the measurement file automatically saves to ensure the data is captured.

## 1.5 Measurement Sequences

A measurement sequence permits the user to construct a number of consecutive sequence steps that will be executed in order. Each sequence step is either an embedded step or a custom scripted measurement. Sequence steps can be installed with MeasureLINK™-MCS software, added as part of an application pack (see section 1.9.1), or registered as a custom step (see section 1.7.7). These sequences permit the user to combine the control of environmental parameters (e.g., temperature or field) with specific measurement routines (e.g., electrical source and measure instruments), and subsequently create tables and charts to review the data. This process provides completely customizable measurements, while hiding the custom script from the user. Edit the UI for each sequence step and then start the sequence.

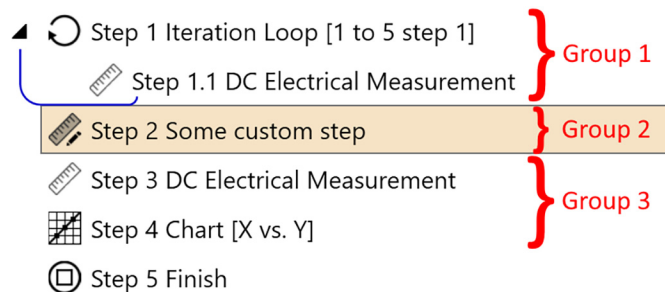
Sequence examples can be found in the MeasureLINK folder on your hard drive: (C:\Users\Public\Documents\MeasureLINK\Examples).

### 1.5.1 Sequence Measurement Concepts

Sequences are managed through the Sequence Measurement window and measurement sequence files (\*.mlseq). A sequence is an ordered series of steps to complete a measurement. The typical sequence workflow consists of adding the desired steps to the sequence, configuring those steps, and then executing the sequence. Any data collected is stored in the sequence file. To protect the data, the File Mode setting (see section 1.5.3.1) can be set to New File Mode. This setting prevents MeasureLINK from overwriting data stored in previously executed measurement sequences..

Any script file (\*.mlscr) can be added to a sequence. When a script file is in a sequence its UI and results will display in the results pane of the Sequence Measurement window. When the script file is added to the sequence, the original script is copied into the sequence. Therefore, any changes made to the script later will NOT be reflected in the sequence. Script files that will be frequently used in a sequence can be registered with MeasureLINK™ and will appear in the list of available sequence steps.

Custom script steps in sequences have some important consequences. Each sequence is divided into groups based on custom steps. Each custom step is its own group, and all groups are independent of each other. For example, see the sequence below:



The loop in Step 1 starts the first group (Group 1), and it contains another sequence step measurement. The custom step at Step 2 forces a new group, Group 2. The following steps (Steps 3-5) are a new group, Group 3. When the sequence starts, each group runs in order, as if they were separate sequences run consecutively. This means custom steps can never interact with normal sequence steps, and that custom steps can isolate sequence steps. Therefore:

- Custom steps can never be inside a sequence loop. For example, the custom Step 2 cannot be put inside the Step 1 iteration loop.
- Steps in a group can only see other steps in that group. For example, the Chart in Step 4 can only chart data from measurements in Group 3. The chart cannot “see” the electrical measurement in Group 1 or any data generated by the custom step.



This restriction is because a custom step does not have to follow any sequence rules and therefore cannot be integrated into a single script with normal sequence steps.

### 1.5.2 Sequence Measurement Window



The controls on the Sequence Measurement window allow you to create and modify a sequence, execute that sequence, and view the results. The images below show a typical sequence.

**All sequences have a Finish step at the end of the sequence. This step cannot be added, moved, or deleted, and marks the end of the sequence.**

- **Sequence ribbon:** contains options for controlling script execution.
- **Step address:** Identifies the position of a step within a sequence.
- **Selected step:** The presently highlighted step. The workspace displays information related to this step.

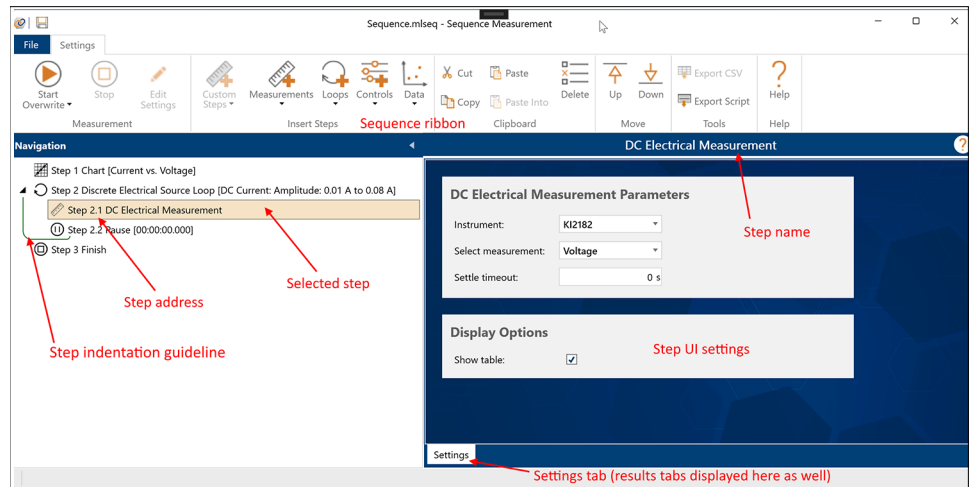


FIGURE 1-6 Sequence before running

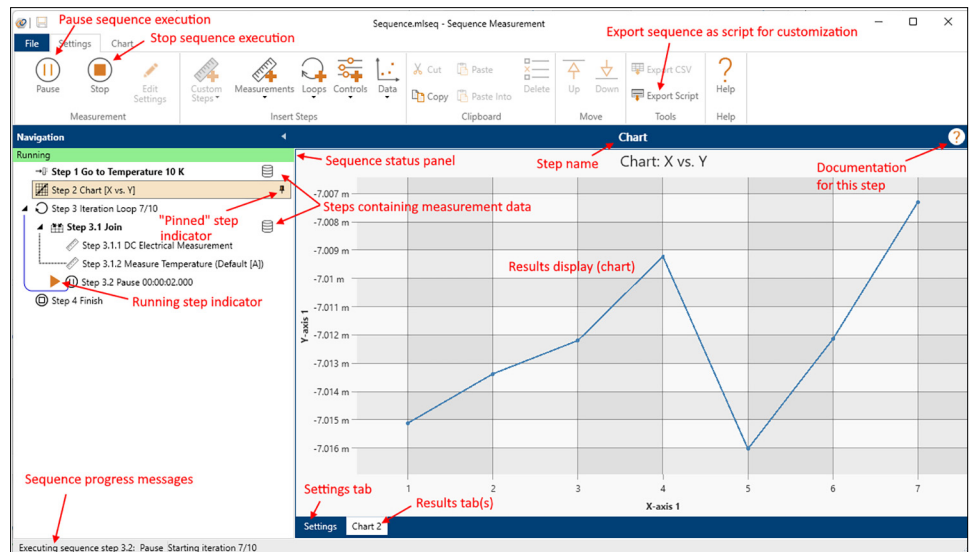


FIGURE 1-7 Sequence while running

- *Navigation pane*: contains a list of all the sequence steps. The current step is indicated here. This area also contains visual indicators of the sequence status and function.
- *Workspace*: the right side of the screen contains a tab control where the software will display settings and results for the current step. The contents of this area vary by the type of step and if the step is displaying results.
- *Data indicators*: any sequence steps that contain measurement data will be displayed in bold and show a data icon on the right side.
- *Pinning*: any sequence step can be "pinned" before or during sequence execution. Pinned steps remain visible until the next pinned step is executed, and some steps are pinned by default. To pin or unpin a step, hover the mouse over the right side of the sequence and click the pin icon.
- *Running step indicator*: indicates the step currently being executed in a running sequence.
- *Loop indicators*: loops allow repeated measurements (see section 1.6.2). Steps inside a loop are indented. In addition, a solid curved line from the last step in the loop to the new loop indicates what is contained in that loop.



**Loops can be collapsed to simplify the view of the sequence.**

- *Group indicators*: the Group and Join steps (see section 1.6.3.4 and section 1.6.3.5) organize sequence steps into logical groups. These look much like loops but are indicated by a angular dashed line.
- *Sequence status panel*: during and after sequence execution, the area directly above the sequence steps will show the sequence execution status using text and color.

## 1.5.3 Sequence Measurement Controls

### 1.5.3.1 Measurement

The Measurement group on the ribbon executes and stops the measurement.

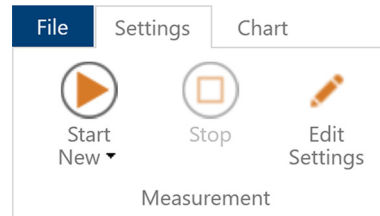


FIGURE 1-8 Measurement

#### 1.5.3.1.1 Start

Click the **Start** button to begin executing the sequence. The **Start** button will be disabled if the sequence has errors, or if another measurement is running.

While a sequence is executing, all measurement data generated by each of the steps in the sequence is written to a temporary location on the computer hard drive. When the sequence completes execution, the data is automatically saved in a sequence file (\*.mlseq). However, the way this is completed varies depending upon which file mode option is selected from the **Start** button drop down list: New File Mode (default), or Overwrite File Mode.

Selecting either of these modes does not execute the measurement, but instead simply sets the desired file mode user preference as indicated by the check mark. Once the desired mode is set, it remains the selected mode until changed by the user, even after MeasureLINK has been closed and restarted.

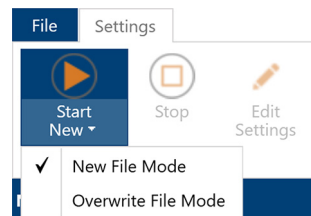


FIGURE 1-9 Start button drop down list

If the **Start** button file mode option is set to New File Mode, the measurement settings and data will be saved to a new sequence file using a unique file name that is automatically generated by MeasureLINK. This new file name is generated if the current file name corresponds to a saved file that already contains measurement data. In this way, no previous measurement data is overwritten. When the file mode option is set to New File Mode, the **Start New** button will display, as shown below.

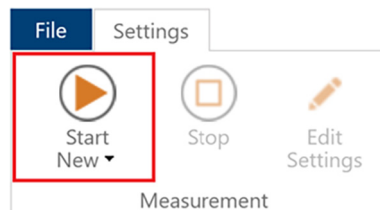


FIGURE 1-10 Start New button

If the **Start** button file mode option is set to **Overwrite File Mode**, the current sequence file will be overwritten with the new measurement data when the sequence completes execution. When the file mode option is set to **Overwrite File Mode**, the **Start Overwrite** button will display, as shown below.

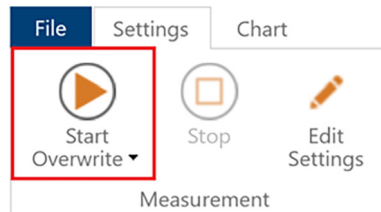


FIGURE 1-11 *Start Overwrite button*

When the **Run** button is clicked while the **Start** button file mode option is set to **Overwrite File Mode**, a warning dialog box will be displayed to remind the user that existing data will be overwritten. This dialog box can be disabled if desired. If desired, previously disabled warning message dialogs can be re-enabled from the **User Preferences** pane (see FIGURE 1-74).

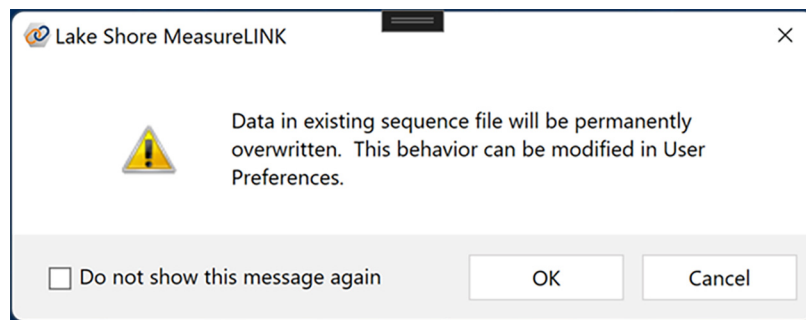


FIGURE 1-12 *Overwrite warning*

### NOTE

**Disabling this dialog box increases the risk of unintentional data loss.**

The **Start** button file mode gives the user the ability to prioritize either the protection of measurement data from the potential risk of unintentional loss, or the convenience of making multiple measurements in the same sequence window without managing multiple sequence files.

### NOTE

**If the automatic save of sequence measurement data at the completion of execution fails for any reason, a dialog box will be displayed alerting the user, describing the error, and asking the user to save the data manually using **Save** or **Save As** in the **Backstage** view.**

A change in the state of measurement data in an open sequence file is not considered when determining if a sequence file that is being closed has been modified (a state often referred to as being “dirty”). If a sequence file that contains data is opened, and then an action that results in the data being cleared from the sequence is performed, a warning to save the sequence before closing will not be displayed, and the file on the disk will not be modified. If the sequence is then re-opened, the data will once again be present.

### 1.5.3.1.2 Pause

While the sequence is running the **Start** button will change to a **Pause** button.

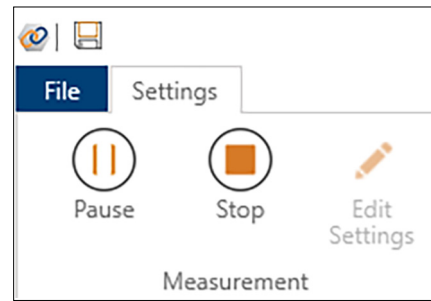


FIGURE 1-13 *Pause button*

Press this button to suspend sequence execution at any time. While paused, the button will change to a **Resume** button.

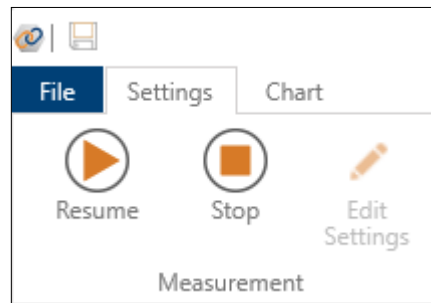


FIGURE 1-14 *Resume button*

**The instrument and drivers will continue to operate when the sequence is paused and the monitor window will continue to update, but no data is collected while the sequence is paused.**

For example, if a sequence is paused inside a Continuous Temperature Loop step, the temperature ramp is still under the control of the selected temperature controller driver. So the temperature will continue to ramp while the sequence is paused. If the loop final temperature is reached, the temperature will settle and the ramp will end.

### 1.5.3.1.3 Stop

While the sequence is running, use the **Stop** button to stop the measurement.



**When a measurement is stopped before it has been completed, the measurement data that is saved to the sequence file may be incomplete.**

### 1.5.3.1.4 Edit Settings

Click the **Edit Settings** button to re-enable the settings for editing in an open sequence that contains measurement data from a previous run.

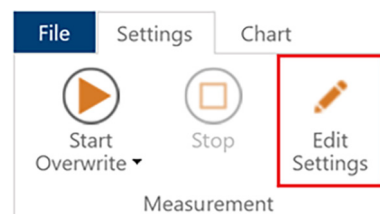


FIGURE 1-15 *Edit Settings button*

Clicking the **Edit Settings** button also clears existing measurement results from the sequence. This ensures that the sequence setup and the sequence results that correspond to that setup, both stored in the sequence file, always remain in sync with each other. To modify the sequence in any way, the existing results need to first be cleared so that the sequence configuration user interface can be re-enabled for editing.

- If the **Start** button file mode is set to **New File Mode**, a new file name will automatically be generated and displayed at the top of the sequence window.
- If the **Start** button file mode is set to **Overwrite File Mode** when the **Edit Settings** button is pressed, a warning dialog box will be displayed because existing data is cleared. The dialog box can be disabled if desired. Previously disabled warning message dialogs can be re-enabled from the **User Preferences** pane (see FIGURE 1-74).

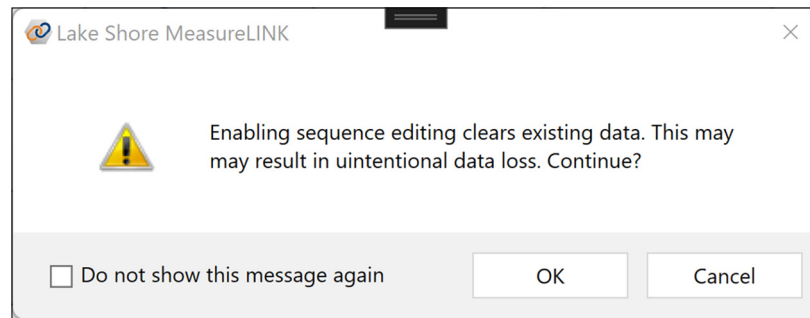


FIGURE 1-16 Reset sequence warning

### 1.5.3.2 Insert Steps

The **Insert Steps** group on the ribbon contains drop down controls to add steps to the sequence.

- **Custom Steps:** contains all the registered scripts (see section 1.7.7). It also contains a **Browse** option to select and import any script file into the sequence.
- **Measurements:** contains steps that collect data from instruments. The actual contents depend on which application packs are installed.
- **Loops:** contains sequence steps for completing different kinds of loops. For example, the **Iteration Loop** repeats all the steps inside the loop a fixed number of times. Depending on the installed application packs, other loops may be installed.
- **Controls:** these steps make changes to the measurement, or the instruments used in the measurements. For example, the **Pause** step simply waits for a certain amount of time to either slow down a measurement or allow a system to settle. Other control sequence steps may be installed by application packs
- **Data:** these steps present data collected during the measurement. Currently the only data sequence step is the **Chart** step (see section 1.9.4.2.3). This step can display data collected in another measurement step as a chart.



**When a script is added to a measurement sequence as a custom step, the script is copied into the sequence. Changes in the original script file will not change the step in the sequence. Currently, the only way to update a script in a sequence is to remove the step and add the modified script as a new step.**

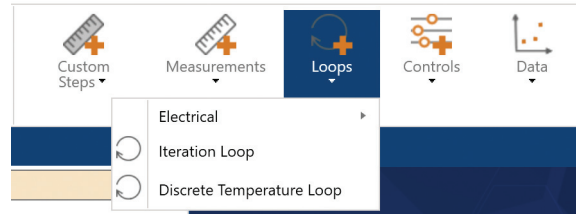


FIGURE 1-17 *Insert Steps*

### 1.5.3.3 Clipboard

The Clipboard group on the ribbon allows clipboard operations on the selected sequence step. A step copied into the clipboard can be pasted anywhere in any sequence. Standard keyboard shortcuts are supported.

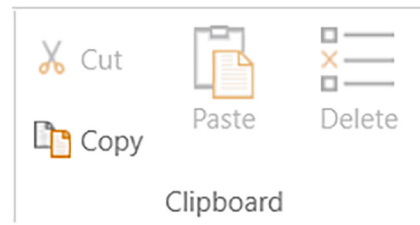


FIGURE 1-18 *Clipboard*

### 1.5.3.4 Move

The buttons in the Move group shift the current step up or down one step in the sequence. No step can be moved after the Finish step.

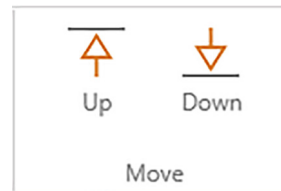


FIGURE 1-19 *Move Up and Move Down buttons*

### 1.5.3.5 Tools

If the current step contains measurement results, use the **Export CSV** button to export data to a comma-separated-value file. If the step contains multiple results, you will be prompted to select the results you wish to export. Then choose a file name.



FIGURE 1-20 *Tools*

## 1.6 Types of Sequence Steps

This section describes the sequence steps that are included by default in MeasureLINK™-MCS software. Other sequence steps may be found based on the application packs (see section 1.9.1) that are installed. Application pack sequence steps are documented in separate help files, available from the Help page of the Backstage view (see section 1.3.6), or by using the Help button in the sequence step ribbon.

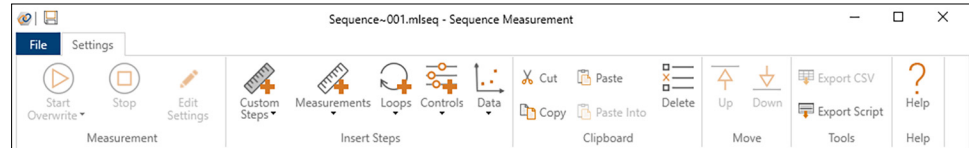


FIGURE 1-21 Sequence ribbon with Help button, right

Use the guidelines and concepts in section 1.5 to complete measurements as listed in the sections below. Add steps to the sequence using the drop down lists under Insert Steps on the Sequence ribbon.

### 1.6.1 Measurements

Add a measurement to a sequence using the Measurements menu on the Sequence ribbon.

#### 1.6.1.1 Measure Temperature Sequence Step

The Measure Temperature step captures a single temperature reading from any temperature controller in the system configuration. By putting this step in a loop, you can capture multiple temperatures. This step can only capture the primary sensor temperature controller. This is usually the sample temperature.

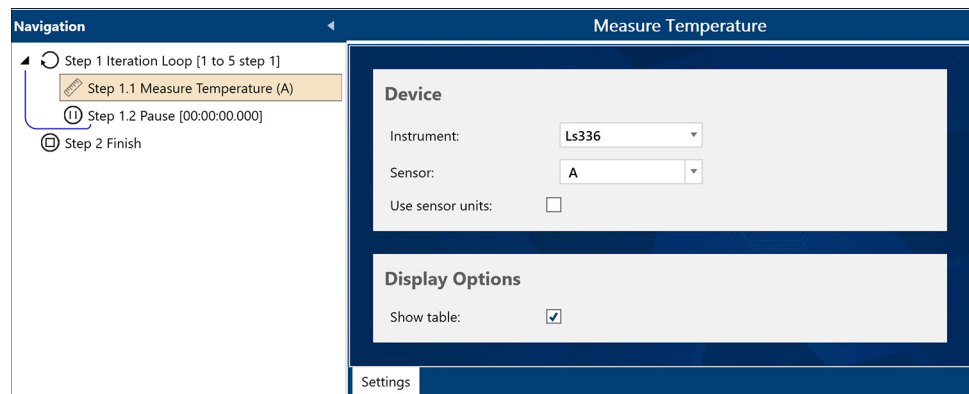


FIGURE 1-22 Measure Temperature

The controls are arranged into two groups:

##### 1.6.1.1.1 Device

The Instrument field in this group contains a list of all the temperature control instruments in the active system configuration. Select the instrument that you wish to use to control the temperature.

- **Sensor:** the selected sensor channel is the channel used to measure during the measurement step. Use the Default option to use the default channel of the selected instrument.
- **Use sensor units:** check this box to use sensor units in the readings taken from the measurement step.

##### 1.6.1.1.2 Display Options

- **Show table:** check this box to display a table of the temperature readings taken by the measurement step



## 1.6.2 Loops

Loops allow measurements to be repeated. Most loops vary some environmental factor, such as field or temperature. Data from the loops is automatically added to the measurement results of the measurements within the loop. For example, a measurement within a temperature loop will have the set and measured temperatures added as additional columns in the measurement results.

Loops can be nested to any level. The grouping of loops steps within the loops is shown in the sequence window by the indent level of the sequence steps and the numbering of the sequence steps. In the example below, Iteration Loop 1 contains Discrete Temperature Loop 1.1, which in turn contains DC Electrical Measurement 1.1.1.

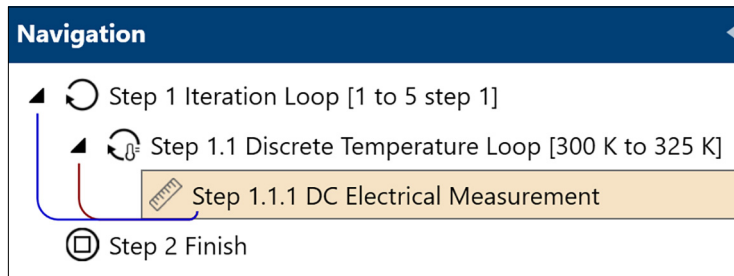


FIGURE 1-23 Nested loops

### 1.6.2.1 Iteration Loop Sequence Step

The Iteration Loop repeats the measurements inside the loop a specified number of times.

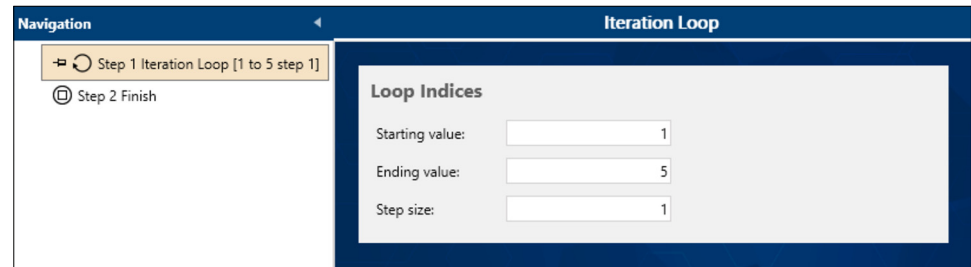


FIGURE 1-24 Iteration Loop

- **Starting value:** initial value of the loop counter. The default is 1; any value is valid.
- **Ending value:** final value of the loop counter. The default is 5, but can be set to any value.
- **Step size:** increment used for the loop counter. The default is 1 and will usually remain at that value. The loop will only perform whole steps. For example, a loop from 1 to 6 with a step size of 2 will do three steps at 1, 3, and 5. Since the difference between 5 and 6 is less than the step size, the loop stops at 5. Also be aware that if the starting value is greater than the ending value, the step size must be negative.

In the sequence, the loop will start at the starting value. Each time through the loop, the value will change by the step size until the loop has reached the ending value. The value of the loop is a counter that is stored in the contained measurements. Decimal (double) values are supported, so as an example, a loop from .2 to 10.5 with an increment of .4 is perfectly valid.

### 1.6.2.2 Continuous Field Loop Sequence Step

The Continuous Field Loop sequence step allows a series of measurements to be performed while field is ramped continuously from a starting value to an ending value. This loop works with any field control instrument, such as the Lake Shore F71 or F41 teslameter.

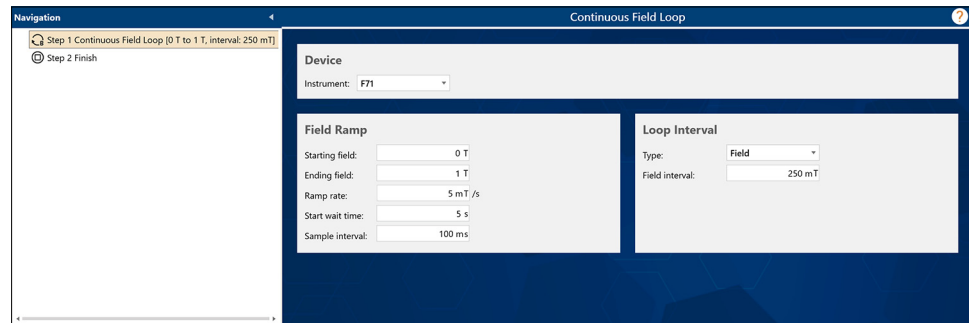


FIGURE 1-25 Continuous Field Loop

The controls are arranged into three groups:

#### 1.6.2.2.1 Device

The Instrument field in this group contains a list of all the field control instruments in the active system configuration. Select the instrument that you wish to use to control the field.

#### 1.6.2.2.2 Field Ramp

The parameters in this group are used to configure the field ramp for the loop:

- **Starting field:** the initial field of the ramp. When loop execution begins, the field controller ramps to this field and settles. This is the first field value at which a measurement is executed. The Starting field and the Ending field must not be equal.
- **Ending field:** the field at which the ramp terminates. When the ramp terminates, loop iterations are also terminated. The loop may or may not execute a measurement at this field, depending upon the loop configuration. The Starting field and the Ending field must not be equal.
- **Ramp rate:** the rate, in field units per second, at which the field controller ramps the field from the Starting field to the Ending field. Set this parameter to a value that is appropriate for the selected field controller configuration. The ramp rate must be a positive value.
- **Start wait time:** additional delay time after the field is settled at the Starting field, and before the ramp to the Ending field begins. This can be set to 0 seconds if no delay is desired. This value must be between 0 and 60 seconds.
- **Sample interval:** the Time interval at which the loop queries the field controller for the current field value during the ramp. These field values are used to determine when to trigger loop measurements when the interval type is set to Field. The smaller this time increment, the more frequently that communication with the field controller takes place. A larger increment may be appropriate when the ramp rate is relatively small, or when the Field interval is relatively large. If this parameter is set to a value that is too large relative to the defined Ramp rate and Field interval, the field loop setpoint values and spacing and may be affected. This value must be between 0.05 and 60 seconds.

#### 1.6.2.2.3 Loop Interval

The parameters in this group are used to configure the looping interval:

- **Type:** the type of Field interval that should be used.
  - **Field:** the default setting. When this interval type is used, the Continuous Field Loop step continually monitors the field ramp, and triggers a loop iteration

each time that the field changes by the amount specified in the Field interval parameter.

- **Time:** when this interval type is used, the Continuous Field Loop step initiates a loop iteration each time that the amount of time specified in the Time interval parameter has expired.
- **Field interval:** the change in field required to trigger each loop iteration. When the field ramp is initiated, the field setpoint for the first loop iteration is set to the Starting field value defined above. For each subsequent loop iteration, the setpoint is calculated by adding the Field interval to the previous loop iteration setpoint. A measured field value for each loop iteration is queried from the field controller at the beginning of each loop iteration. Both the setpoint and measured value are listed in the measurement results tables for measurement steps used inside of field loops. If the field changes by an amount that is more than the defined Field interval during a loop iteration, the next iteration is triggered immediately. The Field interval must be greater than or equal to 0.0. When set to 0.0, loop iterations are triggered immediately after the previous iteration has ended.
- **Time interval:** the change in time required to trigger each loop iteration. When the field ramp is initiated, the field setpoint for the first loop iteration is set to the Starting field value define above. For each subsequent loop iteration, both the setpoint and the measured field value are queried from the field controller at the start of the iteration. For time intervals, these values are identical. Both the setpoint and measured value are listed in the measurement results tables for measurement steps used inside of field loops. If the time changes by an amount that is more than the defined Time interval during a loop iteration, the next iteration is triggered immediately. The Time interval must be between 0.0 and 3600. When set to 0.0, loop iterations are triggered immediately after the previous iteration has ended.

#### 1.6.2.2.4 End of the field loop

When the field ramp reaches the Ending field value and settles, the Continuous Field Loop stops performing loop iterations and the loop step is considered complete.

#### 1.6.2.3 Discrete Field Loop Sequence Step

The Discrete Field Loop sequence step allows measurements to be performed in a multi-segment field loop. This loop works with any field control instrument, such as the Lake Shore F71 or F41 teslameter.

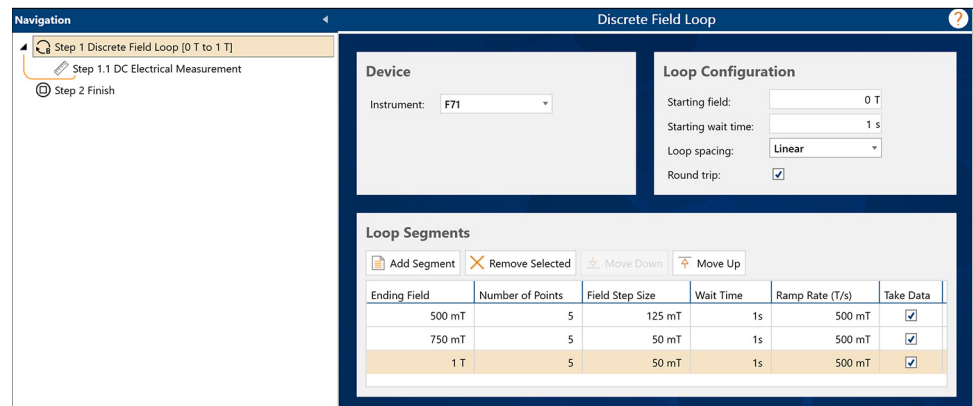


FIGURE 1-26 Discrete Field Loop

The controls are arranged into three groups:

### 1.6.2.3.1 Device

The Instrument field in this group contains all of the field control instruments in the active system configuration. Select the instrument that you wish to use to control the field.

### 1.6.2.3.2 Loop Configuration

This group configures general loop starting conditions and the type of loop:

- **Starting field:** the initial field to start the loop. When loop execution begins the field controller will ramp to this field and settle. This will also be the first field point in the loop.
- **Starting wait time:** additional delay time after the field as settled at the Starting Field. This can be 0 for no delay.
- **Loop Spacing:** three spacings are supported:

Element	Description
<b>Linear</b>	Default setting. Each step in the field loop is the same size as the previous step.
<b>Logarithmic</b>	Step sizes change in a logarithmic progression. Due to the properties of this function, the starting and ending points of the progression must be the same sign, and neither can be 0.
<b>Inverse parameter</b>	Step sizes are based on 1/B. Due to the properties of this function, the field can never be 0.

- **Round trip:** when checked, the loop will return to the starting field, visiting all the same setpoints on the return branch of the loop as it did on the initial branch of the loop. This doubles the number of points in the loop.

### 1.6.2.3.3 Loop Segments

The field loop may have multiple segments. These segments are defined in the Loop Segments group. Any number of segments can be added, removed, or moved using the buttons above the segment grid. Click a value to edit it.

#### Segment parameters

For each segment, the following values must be provided. All field values are in your preferred field units.

Parameter	Description
<b>Ending Field</b>	The field that marks the end of the segment. See <b>Field loop step calculations</b> , below, for more information on how the loop is calculated.
<b>Number of Points</b>	The number of field points to visit in this segment.
<b>Field Step Size</b>	This read-only value is only displayed for linear loops. It reflects the field step spacing for the segment and is calculated based on the Ending Field and the Number of Points.
<b>Wait Time</b>	Optional additional wait time, in seconds, after the field settles at each point in the segment.
<b>Ramp Rate</b>	Ramp rate to use for this segment in preferred field units per second. Though the ramp rate is initially populated with a default value, t enter a value that is appropriate for the field platform being used.
<b>Take Data</b>	If checked, (the default) data will be collected for each point in this segment. If unchecked, the field will ramp to each point in the segment but no measurements inside the loop will be performed.

#### Segment rules

The only rule for linear loops is that each segment must have at least two points. Other loop spacings follow other rules.

#### Field loop step calculations

- The first segment runs from the loop starting field through the segment ending field.

- Subsequent segments run from the previous segment’s ending field to the segment ending field. If a round trip is selected, the final segment’s ending field will be the starting field for the return leg of the round trip. This algorithm has two side effects:
  1. The point spacing calculation is slightly different for the first segment because the starting field is included as the first point. For even step sizes across segments, add an additional point to the first segment.
  2. On round trip loops, the ending field of the last segment is only visited once. This causes the descending segments to shift by one step, and the return leg of the first segment to have one less data point than the ascending leg.

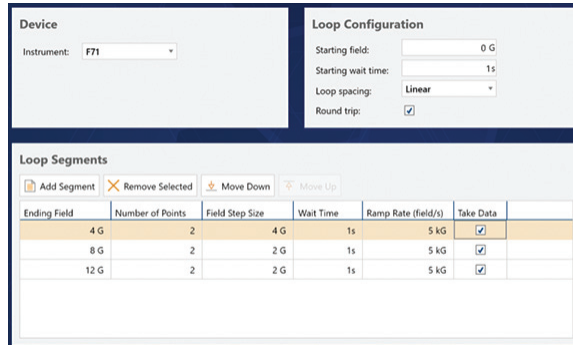
The following figure is an example of a field loop step calculation, resulting in the following fields being visited in the order shown:

First “leg” of the round trip:

- Segment 1: 0, 4
- Segment 2: 6, 8
- Segment 3: 10, 12

Return “leg” of the round trip:

- Segment 3: 10, 9
- Segment 2: 6, 4
- Segment 1: 0



#### 1.6.2.3.4 End of the field loop

When the last segment is complete, the field is left under control at the last field setpoint visited.

#### 1.6.2.4 Continuous Temperature Loop Sequence Step

The Continuous Temperature Loop sequence allows a series of measurements to be performed while temperature is ramped continuously from a starting value to an ending value. This loop works with any temperature control instrument, such as the Lake Shore 335 or 336 temperature controller.

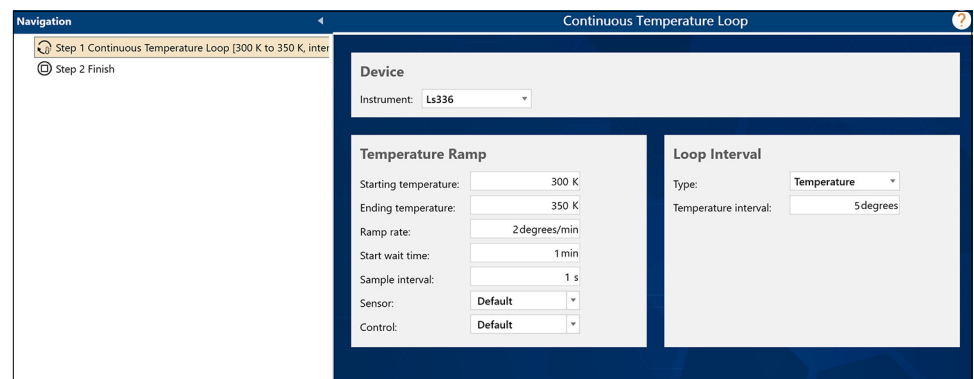


FIGURE 1-27 Continuous Temperature Loop

The controls are arranged into three groups:

#### 1.6.2.4.1 Device

The Instrument temperature in this group contains a list of all the temperature control instruments in the active system configuration. Select the instrument that you wish to use to control the temperature.

#### 1.6.2.4.2 Temperature Ramp

The parameters in this group are used to configure the temperature ramp for the loop:

- **Starting temperature:** the initial temperature of the ramp. When loop execution begins, the temperature controller ramps to this temperature. This is the first temperature value at which a measurement is executed. The Starting temperature and the Ending temperature must not be equal.
- **Ending temperature:** the temperature at which the ramp terminates. When the ramp terminates, loop iterations are also terminated. The loop may or may not execute a measurement at this temperature, depending upon the loop configuration. The Starting temperature and the Ending temperature must not be equal.
- **Ramp rate:** the rate, in degrees (C or K) per minute, at which the temperature controller ramps the temperature from the Starting temperature to the Ending temperature. Set this parameter to a value that is appropriate for the selected temperature controller configuration. The ramp rate must be a positive value.
- **Start wait time:** additional delay time after the temperature is settled at the Starting temperature, and before the ramp to the Ending temperature begins. This can be set to 0 minutes if no delay is desired. This value must be between 0 and 60 minutes.
- **Sample interval:** the Time interval at which the loop queries the temperature controller for the current temperature value during the ramp. These temperature values are used to determine when to trigger loop measurements when the interval type is set to Temperature. The smaller this time increment, the more frequently that communication with the temperature controller takes place. A larger increment may be appropriate when the ramp rate is relatively small, or when the Temperature interval is relatively large. If this parameter is set to a value that is too large relative to the defined ramp rate and Temperature interval, the temperature loop setpoint values and spacing and may be affected. This value must be between 0.05 and 60 seconds.

#### 1.6.2.4.3 Loop Interval

The parameters in this group are used to configure the looping interval:

- **Type:** the type of Temperature interval that should be used.
  - **Temperature:** the default setting. When this interval type is used, the Continuous Temperature Loop step continually monitors the temperature ramp and triggers a loop iteration each time the temperature changes by the amount specified in the Temperature interval parameter.
  - **Time:** when this interval type is used, the Continuous Temperature Loop step initiates a loop iteration each time that the amount of time specified in the Time interval parameter has expired.
- **Temperature interval:** the change in temperature required to trigger each loop iteration. When the temperature ramp is initiated, the temperature setpoint for the first loop iteration is set to the Starting temperature value defined above. For each subsequent loop iteration, the setpoint is calculated by adding the Temperature interval to the previous loop iteration setpoint. Additionally, a measured temperature value for each loop iteration is queried from the temperature controller at the beginning of each loop iteration. Both the setpoint and measured value are listed in the measurement results tables for measurement steps that used inside of temperature loops. If the temperature changes by an amount that is more than the defined Temperature interval during a loop iteration, the next iteration is triggered immediately. The Temperature interval must be

greater than or equal to 0.0. When set to 0.0, loop iterations are triggered immediately after the previous iteration has ended.

#### 1.6.2.4.4 End of the temperature loop

When the temperature ramp reaches the Ending temperature value and settles, the Continuous Temperature Loop stops performing loop iterations and the loop step is considered complete.

#### 1.6.2.5 Discrete Temperature Loop Sequence Step

The Discrete Temperature Loop sequence step allows a series of measurements to be performed while temperature is ramped in steps from a starting value to an ending value. This loop works with any temperature control instrument, such as the Lake Shore 335 or 336 temperature controller.

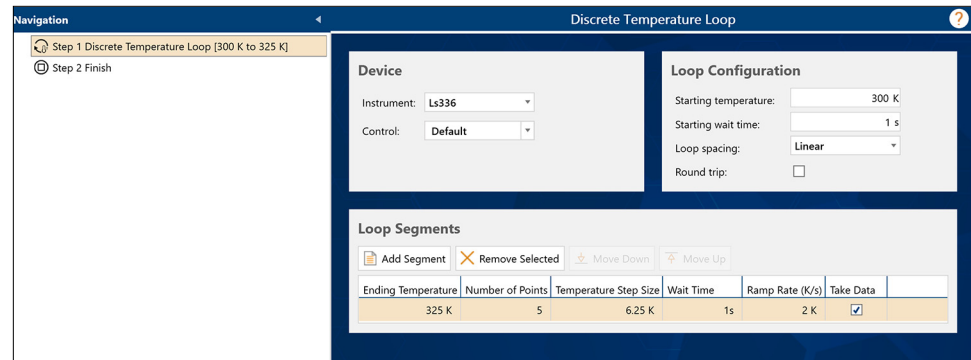


FIGURE 1-28 Discrete Temperature Loop

##### 1.6.2.5.1 Device

- **Instrument:** contains a list of all the temperature control instruments in the active system configuration. Select the instrument to use to control the temperature.

##### 1.6.2.5.2 Loop Configuration

These controls define the temperature loop:

- **Starting temperature:** the initial temperature of the ramp. When loop execution begins, the temperature controller ramps to this temperature. This is the first temperature value at which measurements inside the loop are performed. The Starting temperature and the Ending temperature must not be equal.
- **Starting wait time:** the time to wait before initiating the first segment of the temperature loop.
- **Loop spacing:** three spacings are supported:

Element	Description
<b>Linear</b>	Default setting. Each step in the temperature loop is the same size as the previous step.
<b>Logarithmic</b>	Step sizes change in a logarithmic progression. Due to the properties of this function, the starting and ending points of the progression must be the same sign.
<b>Inverse parameter</b>	Step sizes are based on 1/B.

- **Round trip:** when checked, the loop will return to the starting temperature, visiting all the same setpoints on the return branch of the loop as it did on the initial branch of the loop. This doubles the number of points in the loop.

##### 1.6.2.5.3 Loop Segments

The first segment runs from the loop starting temperature through the segment ending temperature.

Subsequent segments run from the previous segment's ending temperature to the segment ending temperature. If a round trip is selected, the final segment's ending temperature will be the starting temperature for the return leg of the round trip. This algorithm has two side effects:

- The point spacing calculation is slightly different for the first segment because the starting temperature is included as the first point. For even step sizes across segments, add an additional point to the first segment.
- On round trip loops, the ending temperature of the last segment is only visited once. This causes the descending segments to shift by one step, and the return leg of the first segment to have one less data point than the ascending leg.

## 1.6.3 Controls

Add a control to a sequence using the Controls menu on the Sequence ribbon.

### 1.6.3.1 Go to Field with Chart Sequence Step

This sequence step ramps any field controller to a new magnetic field using the provided setpoint. Data collected during the ramp can be optionally shown on a chart.

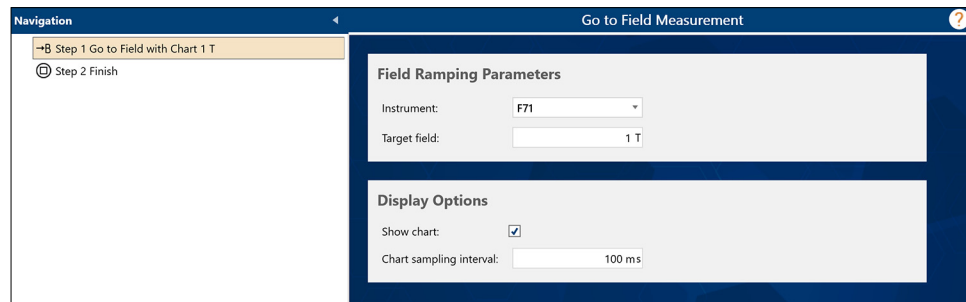


FIGURE 1-29 Go to Field with Chart

The controls are arranged into two groups:

#### 1.6.3.1.1 Field Ramping Parameters

Use these fields to define how you want the field ramp to behave:

- **Instrument:** contains all the valid field control instruments in the active system configuration. Select the instrument that you wish to use to control the magnetic field.
- **Target field:** target field setpoint for the field ramp. If this is higher or lower than the field platform can achieve, the step may never complete the ramp.
- **Use default ramp rate:** if checked, the platform will ramp at its default ramp rate; otherwise, the user-defined ramp rate will be used. This value is in degrees per minute.
- **Ramp rate:** user-defined ramp rate that can be set if Use default ramp rate is unchecked. If the value provided is beyond the capabilities of the temperature platform, you may not get the desired ramp rate.
- **Extra settle time:** additional time in seconds to wait after the temperature ramp is completed. Data will continue to be collected during this time. Use this option to ensure that the temperature is completely settled at the new setpoint.

#### 1.6.3.1.2 Display Options

Use the options in this group to define how the collected data is displayed:

- **Show chart:** check this box to display a table of the temperature readings taken by the measurement step
- **Chart sampling interval:** specifies how often data should be collected. The value is in milliseconds. For slow systems, this can be a longer interval.



### 1.6.3.2 Go to Temperature with Chart Sequence Step

This sequence step ramps any temperature controller to a new temperature using the provided setpoint and ramp rate. Data is collected during the ramp can be optionally shown on a chart.

The screenshot shows a software interface for configuring a temperature ramping step. On the left is a navigation pane with two steps: 'Step 1 Go to Temperature with Chart 300 K' (selected) and 'Step 2 Finish'. The main area is titled 'Go to Temperature with Chart 300 K' and contains two sections:

- Temperature Ramping Parameters:**
  - Instrument: Ls336 (dropdown)
  - Note: *Note that ramp rate will never exceed the maximum rate of the platform and that the platform may not be able to reach the target temperature.*
  - Target temperature: 300 K (input)
  - Use default ramp rate:
  - Ramp rate: 2degrees/min (input)
  - Extra settle time: 0 s (input)
  - Sensor: Default (dropdown)
  - Control: Default (dropdown)
- Data Options:**
  - Show chart:
  - Data sampling interval: 1 s (input)

FIGURE 1-30 *Go to Temperature with Chart*

The controls are arranged into two groups:

#### 1.6.3.2.1 Temperature Ramping Parameters

Use these fields to define how you want the temperature ramp to behave:

- **Instrument:** contains all the valid temperature control instruments in the active system configuration. Select the instrument that you wish to use to control the temperature.
- **Target temperature:** target temperature setpoint for the temperature ramp. If this is higher or lower than the temperature platform can achieve, the step may never complete the ramp.
- **Use default ramp rate:** If checked, the platform will ramp at its default ramp rate, otherwise the user-defined ramp rate will be used. This value is in degrees per minute.
- **Ramp rate:** user-defined ramp rate that can be set if Use default ramp rate is unchecked. If the value provided is beyond the capabilities of the temperature platform, you may not get the desired ramp rate.
- **Extra settle time:** additional time in seconds to wait after the temperature ramp is completed. Data will continue to be collected during this time. Use this to ensure your temperature is completely settled at the new setpoint.

#### 1.6.3.2.2 Data Options

Use the options in this group to define how the collected data is displayed:

- **Show chart:** check this box to display a chart of the temperature vs. time for the duration of the temperature ramp.
- **Data sampling interval:** specifies how often data should be collected. The value is in milliseconds. For slow systems, this can be a longer interval.

### 1.6.3.3 Pause Sequence Step

This sequence step suspends sequence execution for either a fixed time period or until the user elects to continue. No data is collected by this step. For pauses longer than 1 s or while waiting for the user, a dialog window is displayed to show the pause status.



**The pause is only accurate to within a few milliseconds.**

#### 1.6.3.3.1 Pause settings

The pause can be set to wait for a time interval, in which case you can select the hours, minutes, seconds, and milliseconds to wait, or wait for the user signal to continue. In either case, use the Dialog message field to provide custom text to display in the dialog box (see the following section).

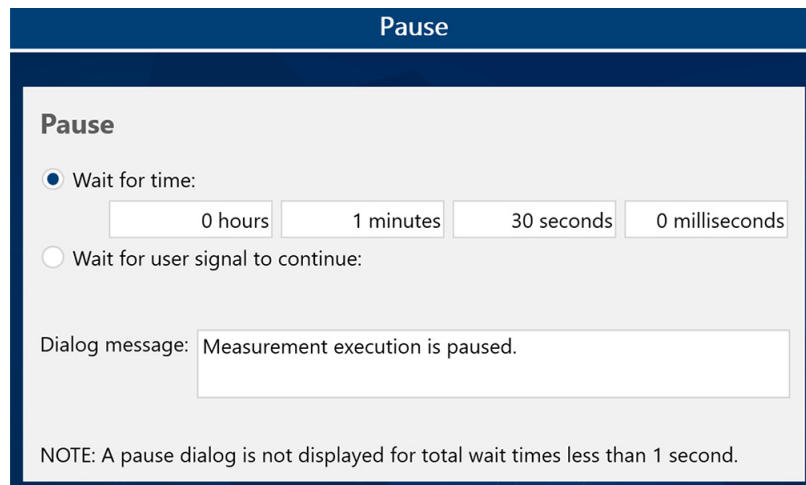


FIGURE 1-31 Pause settings

#### 1.6.3.3.2 Wait for time dialog box

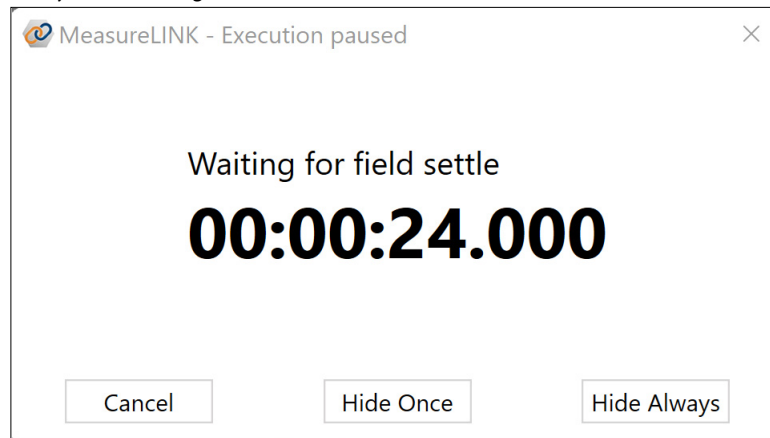


FIGURE 1-32 Wait for time dialog box

For wait times of 1 second or more, this dialog box will pop up to show the timer counting down. It offers three options:

- **Cancel:** abort the wait and continue on with the sequence.
- **Hide Once:** close the dialog box. The sequence will still wait for the proper interval. This is valuable because the dialog can be distracting and also prevents you from navigating to the sequence to look at data. If the pause is inside a loop, the dialog will reappear the next time the pause executes.

- *Hide Always*: works like Hide Once except that even if the pause executes again, the dialog will not re-appear.

Even when the dialog is closed, you can still see the countdown status in the Pause sequences step.

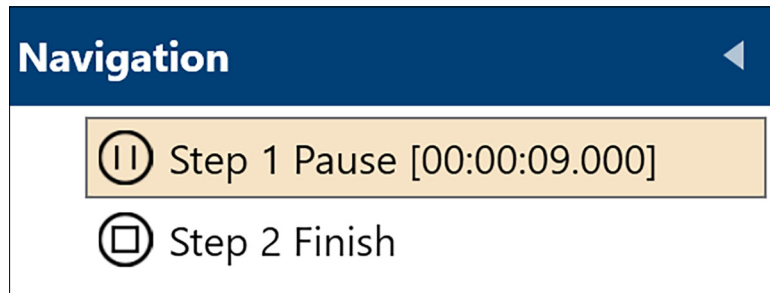


FIGURE 1-33 *Pause sequence countdown*

#### 1.6.3.3.3 Wait for user signal dialog box

When the pause is waiting for the user, the only option is to Continue with the sequence.

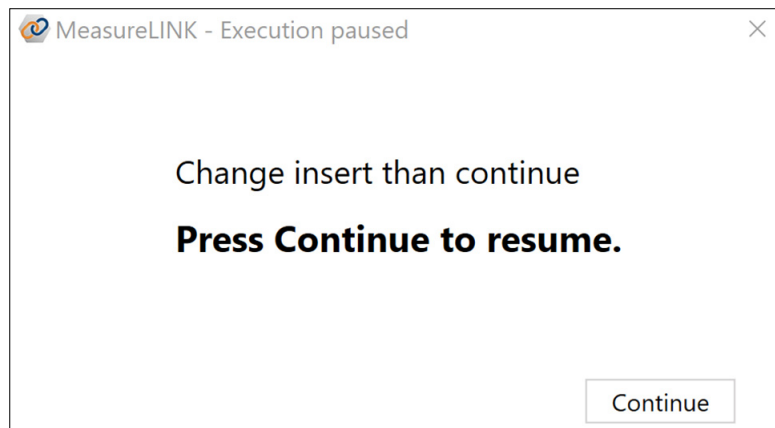


FIGURE 1-34 *Wait for user signal dialog box*

#### 1.6.3.4 Sequence Group Step

The Group step is simply a container to hold other sequence steps. This allows the ability to move several related sequence steps around in the sequence, or to copy and paste the entire group at one time.

Other measurement steps can be placed “inside” a Group just like they can be put in a loop. This group can then be collapsed, moved, copied, or deleted as a single unit in the sequence.

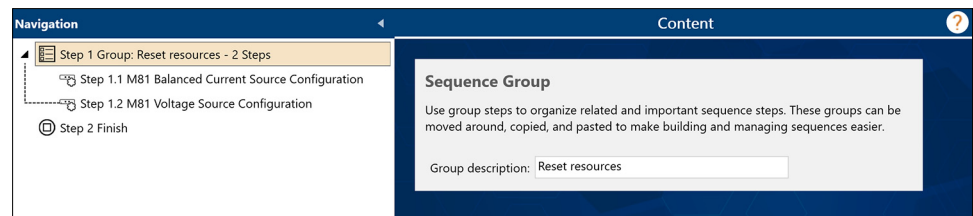


FIGURE 1-35 *Sequence Group step*

The only setting on the Sequence Group step is a text description, which allows the user to assign a descriptive name to the group to help recognize it in the sequence.

### 1.6.3.5 Sequence Join Step

A Join step allows multiple measurement steps to be “joined” into one results table. Normally, each measurement step produces one or more measurement results tables, and each measurement step is isolated from every other measurement step. Within certain limitations, a Join step can combine measurements so they produce a single Measurement Results containing all the data.

For example:

1. This sequence takes an electrical measurement and a temperature measurement every  $\frac{1}{2}$  s, 10 times. To show a graph of voltage over temperature, add a Chart step.

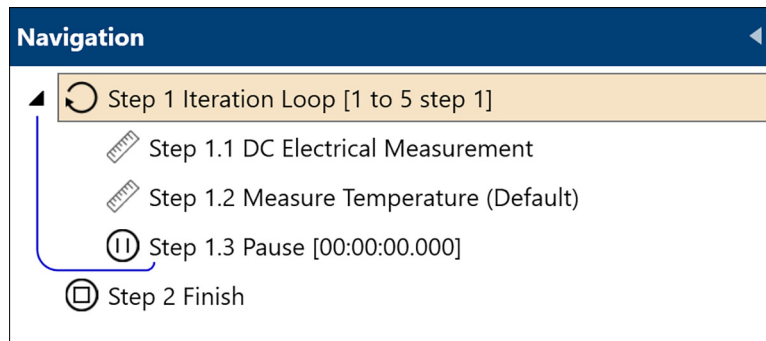


FIGURE 1-36 Join example 1

2. But the results are separate; there is no way to directly chart voltage vs. temperature.

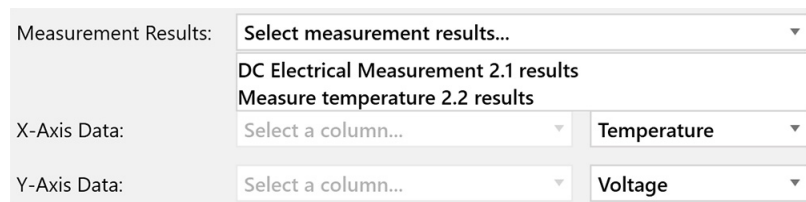


FIGURE 1-37 Join example 2

3. Use a Join step to include all of the measurements, combining the electrical and temperature data into the Join steps Measurement Results.

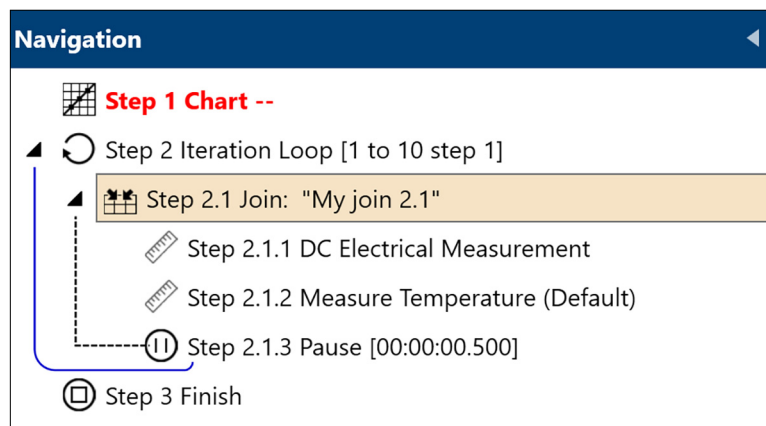


FIGURE 1-38 Join example 3

4. Now the Chart step can directly access temperature and voltage from the Join results.

FIGURE 1-39 Join example 4

#### 1.6.3.5.1 Join settings

There are only two settings for the Join step. The first is the title for the joined data. This allows you to pick a descriptive name. The second is a checkbox that when checked will show all the joined data in a table when the sequence runs.

FIGURE 1-40 Join settings

#### 1.6.3.5.2 Join rules and limitations

Not all measurements can be joined, and there are some other rules to keep in mind when using Join steps:

- Only measurements that produce a single row of data in each measurement can participate in Joins.

If you have Measurement A that produces one row of data, and Measurement B that produces three rows, the software has no way to match the rows. However, if both measurements only produce one row, they naturally align. At this time, these are all the measurements that can be in a Join:

- DC Electrical Measurement
- Temperature Measurement
- M81 Source Measure Synchronized Measurement

### NOTE

**More single-row measurements will be added in the future.**

- Other sequence steps can be included in a Join step (including other Joins), but any measurements will not be included in the Join results. Steps that are not included in the Join will continue to create their own measurement results tables as if the Join did not exist.
- The row of data for a Join is added when the last step in the Join completes. This is not necessarily when the data was collected. For example, if a Join contains two measure steps separated by a one hour Pause step, then the second measure step will do its measurement one hour after the first measurement, but there will still be one row of data added with the time marked as just after the second measurement completed.

### 1.6.3.6 Sequence Finish Step

The Finish sequence step is included in every sequence created within MeasureLINK. It cannot be added, moved, or deleted. The step allows for the setup and management of operations to be performed at the end of a measurement. These operations are performed when a measurement completes execution, is canceled, or stops due to an error. If another measurement is started while finish operations are executing, the operations will be canceled and the new measurement will begin..

Instrument drivers for MeasureLINK that implement support for Finish operations will have dedicated settings. Driver finish operations can be configured from the driver's settings page, or via a sequence configuration step.

The Finish step screen displays the Finish Step Operations, and a Finish Status as they are executed. Once instruments are setup in the system configuration with Finish operations, those operations will be displayed.

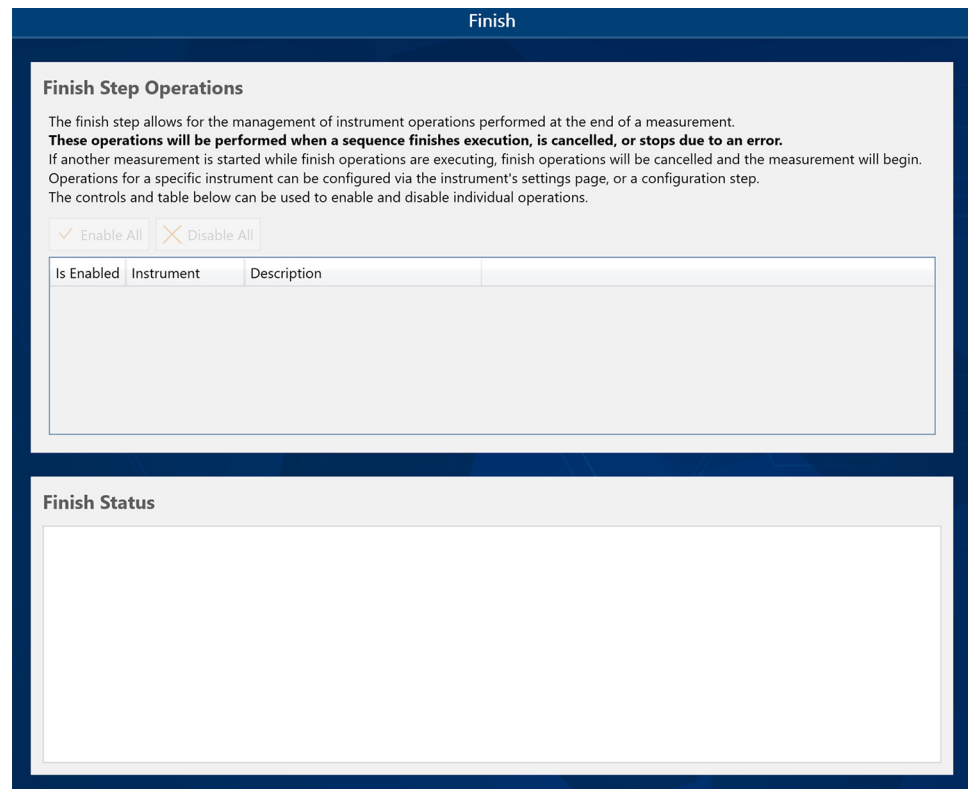


FIGURE 1-41 Sequence Finish settings

Each row of the Finish Step Operations table represents one instrument's set of Finish operations, with the instrument name and a description of the operation. Each row can also be enabled or disabled; disabled operations will not be run. Operations appear in the table in the same order as the system configuration.

At the end of a measurement, as Finish operations execute, status information is relayed to the Finish Status section of the screen. Updates will be published when operations start, end, or result in an error.

## 1.6.4 Data

Add data to a sequence using the Data menu on the Sequence ribbon.

### 1.6.4.1 Chart Sequence Step

The Chart sequence step displays measurement data from your current instruments on one or more charts when running a sequence. All charts have configurable plot settings.

The Chart step has a Properties editor group, Chart tree, and group for results tabs. The properties editor is used to configure different chart elements. The chart tree is used to arrange chart elements and select them for editing. New charts generated during a sequence will be added as a new results tabs.

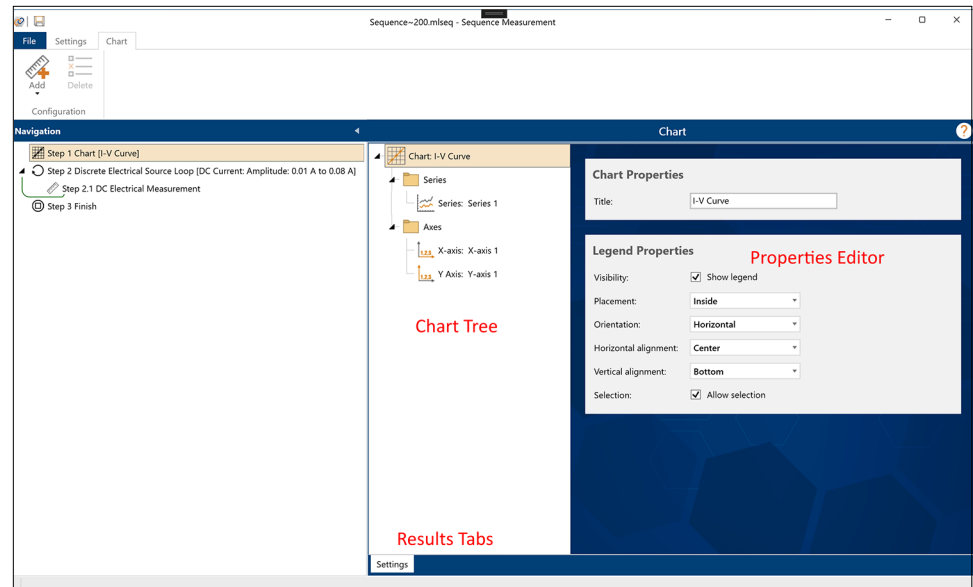


FIGURE 1-42 Chart sequence step

#### 1.6.4.1.1 Chart tree and properties editor

The top of the Chart sequence step is split into two areas. On the left is the chart tree, and on the right is the property editor for the currently selected item in the tree. The chart step uses a hierarchical tree model where each item in the tree “owns” the items below it. For example, the Chart contains a Series group and an Axes group, which each contain the proper components, respectively.

Element	Description
<b>Chart</b>	The root of the tree. Cannot be deleted or added. Contains chart-level properties such as title and legend options.
<b>Series group</b>	Charts can plot multiple series, all of which must be contained in the chart’s Series group.
<b>Axes group</b>	Charts can plot different series on different axes, all of which must be contained in the chart’s Axes group.
<b>Series</b>	Each series displays the data collected in a measurement results object. Series properties include series name, series appearance, and axes configuration. All charts must have at least one series.
<b>X-Axis</b>	Different X-axes can be applied to different series to display data of different types and scales. X-axis properties control the title, data scale, axis position, and data type. All charts must have at least one X-axis.
<b>Y-Axis</b>	Different Y-axes can be applied to different series to display data of different types and scales. Y-axis properties control the title, data scale, axis position, and data type. All charts must have at least one Y-axis.

Chart tree elements are added and deleted using the ribbon controls. For example, to add a new series to the series group, select the series group and use the **Add** button that appears in the ribbon. Each chart step element is described in more detail below.

### 1.6.4.1.2 Chart root element

Chart Properties control how the chart appears.

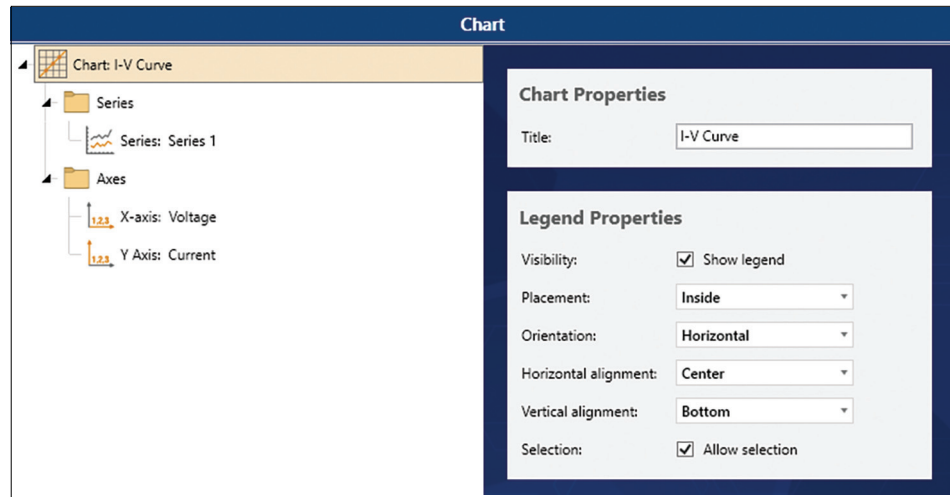


FIGURE 1-43 Chart root element

These controls are arranged into two groups:

- **Chart Properties:**
  - **Chart title:** This title appears on top of the chart. It can be blank or any text you like.
- **Legend Properties:** optionally, the chart can show a legend describing all the series displayed on the chart. Use the controls in this group to control the placement, orientation, and alignment of the legend. If the “Allow Selection” checkbox is selected, each series in the legend will contain a checkbox to allow hiding or showing that series after the data is displayed.

### 1.6.4.1.3 Series group element

The Series Group element does not have properties to set. Instead, it displays a short message about the series on the chart.

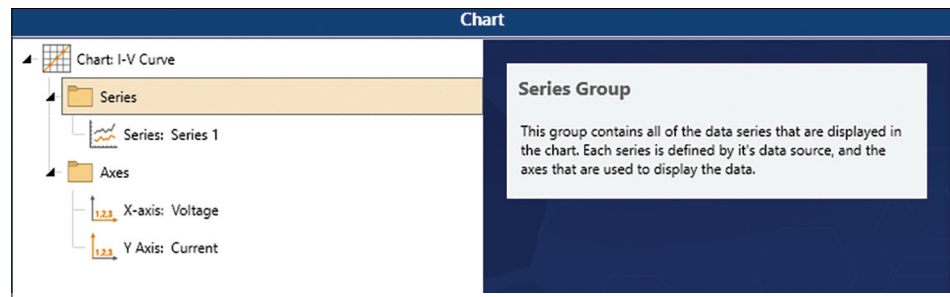


FIGURE 1-44 Series Group element



#### 1.6.4.1.4 Axes group element

The Axes Group element does not have properties to set. Instead, it displays a short message about the axes on the chart.

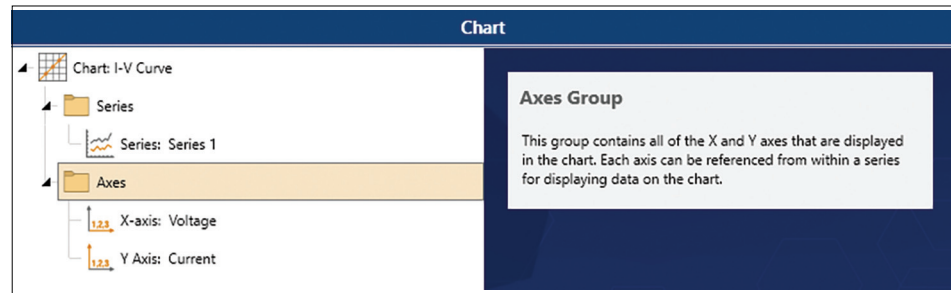


FIGURE 1-45 Axes Group element

#### 1.6.4.1.5 Series element

The series properties control the data in a series and the appearance of series on the chart.

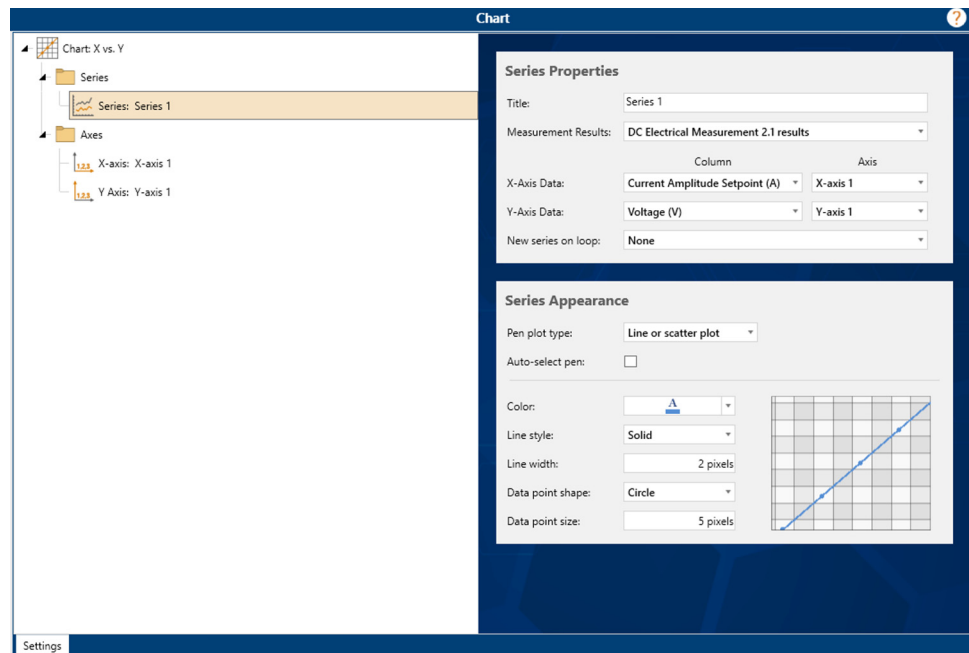


FIGURE 1-46 Series element

- **Series Properties:** Each series on the chart is a sequence of X and Y values. These values can come from any measurement in the sequence, but both the X and Y values must come from the same measurement results.
  - **Title:** the name of the series. The title will appear when hovering the mouse over the data in the chart, and in the chart legend. If the chart produces multiple series, each one will have a number appended to the title.
  - **Measurement Results:** contains a list of the Measurement Results from all the measurements in your sequence. Select the results that contains the data you wish to chart.
  - **Axis Data:** for both X and Y, select the column in the Measurement Results that you wish to use for that value, as well as the axis that the data should be associated with. Column data can only be displayed on axes with the same data type as the column in the Measurement Results.
  - **New series on loop:** when data is being collected in one or more loops, a new series on the chart should be added for each loop. Once a Measurement Result is selected, this control will contain all the loops in which the

measurement that produced the data is contained. Select a loop to start a new series on the chart. If you do not want series breaks or the measurement is not in a loop, then select the default value of None. When the new series is created on the chart, it will be automatically assigned a unique pen.

- **Series Appearance:** The series can display line, scatter, and column (bar) charts. The type and appearance of the plot is determined by the series appearance settings. A sample of how the series will display is shown and will update as the settings change.

Each new series in a chart is assigned a unique pen to draw the series, which is auto-selected by default. The automatic pen selection ensures each series is unique on the chart. Uncheck the auto-select pen to customize the appearance of the series pen.



**If the data is in a loop and “New series on loop” is selected, then only the first pen can be customized. Subsequent loop iterations will auto-select a unique pen.**

All plot types support a color setting. Use the drop down list to select the best color for your series. Other settings change based on the Pen plot type.

- Line or scatter plot settings:
  - *Line style:* selects the way the series line connecting the data points draws. The default is a solid line. Select None to do a scatter plot with no lines.
  - *Line width:* the width of the series line in pixels, from 1 to 15 pixels. The default is 2.
  - *Data point shape:* each data point will be marked with the selected shape. Select None to not display data points.
  - *Data point size:* the width and height of each data point, from 1 to 15 pixels.
- Column or bar plot settings:
  - *Column width percentage:* sets the width of the columns in the series. The value is the percentage of the available space; 100% will have no gaps between the columns, and 50% will have gaps the same width as the columns. Values greater than 100% will cause the columns to overlap. Unlike most settings, this value changes the width of auto-selected pens for the series as well.

### 1.6.4.1.6 X-axis and Y-axis elements

These sections control the appearance and data type of the chart axes. The X-axis is the horizontal axis, and the Y-axis is the vertical axis. The chart can have multiple X and Y axes.

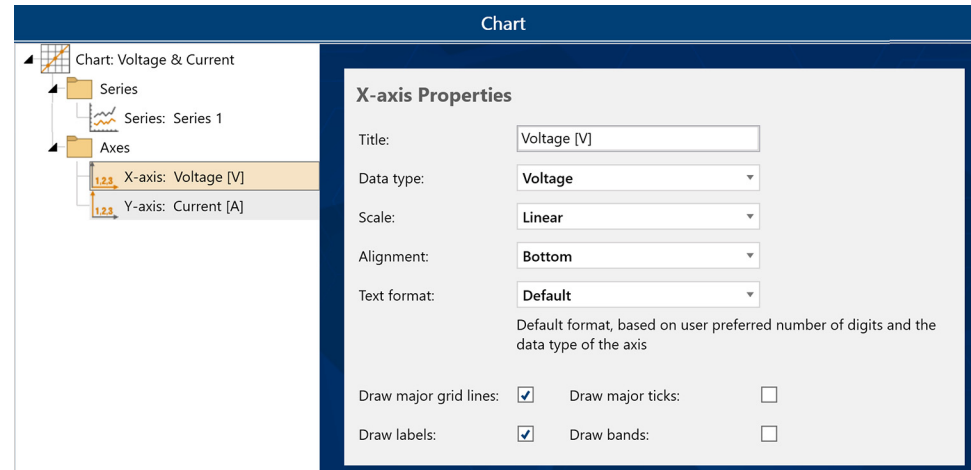


FIGURE 1-47 X-axis and Y-axis elements

Axis properties are similar for X and Y axes, so both are listed below. Any differences are noted.

- **Title:** appears next to the axis on the chart. It can be blank or contain any desired text.
- **Data type:** the axis data type. The following types are supported. Some Data types noted in the table are formatted to the number of significant digits specified for that type in User Preferences (see section 1.8.4). This can be overridden if the Text format is changed.

Data type	Notes
<b>Double</b>	All chartable data <i>except</i> DateTime can map to a double column.
<b>Field</b>	Can only show Field values. Field values will be mapped to the units of the first row in the MeasurementResults.
<b>Integer</b>	“Whole” numbers 0, 1, 2, 3. Can only show single values on an integer axis.
<b>Temperature</b>	Can only show Temperature values. Temperature values will be mapped to the units of the first row in the MeasurementResults.
<b>Voltage</b>	Formatted according to User Preferences by default.
<b>Current</b>	Formatted according to User Preferences by default.
<b>Resistance</b>	Formatted according to User Preferences by default.
<b>Resistivity</b>	Formatted according to User Preferences by default.
<b>Boolean</b>	Maps to 1 or True, 0 for False.
<b>TimeSpan</b>	A time interval value. Only allowed for the X-axis. Can only show TimeSpan data.
<b>DateTime</b>	Date and time data. Only allowed for the X-axis. Can only show DateTime data.

#### NOTE

The Data type of the axis must be compatible with the Data type of the MeasurementResults column to display that axis in the Series. The following rules apply:

- **DateTime** data can only be shown on a DateTime axis, and a DateTime axis can only show DateTime data.
- **A double** axis can show any data that maps to a double. This includes all Data types *except* DateTime.
- **All other axis Data types** can only be mapped to a column of that Data type.

These rules are validated by the Chart Step so they can not be violated.

- **Scale:** how the axis scales data. There are three options: Linear (default), Logarithmic, and 1/X (for the X-axis only). The DateTime axis only supports Linear scaling.
- **Alignment:** determines where on the chart the axis is drawn. Top or Bottom are allowed for an X-axis, Left or Right for a Y-axis.
- **Text format:** lists options for formatting the labels on the axis. These are specific for the Data type of the axis (TimeSpan and DateTime have different format options than the rest). When selected, each format will show a description with details and examples of how the format works.

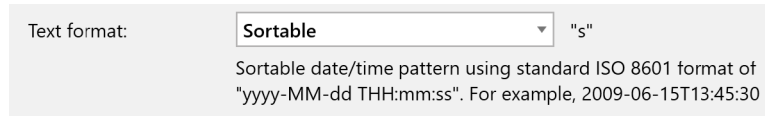


FIGURE 1-48 Text format

Special cases for each Data type are noted below:

- **DateTime:** the default format for DateTimes is “Mixed”. This is a split format that shows a short date format (e.g. “dd/mm/yy hh:mm:ss”) if the axis is showing more than 24 hours, and a time format (“hh:mm:ss”) if the axis shows less than 24 hours.
- **TimeSpan:** due to limitations of our charting solution, most format options are not culture-aware. For instance, the separator may always be ‘:’, even if the computer language settings specify ‘.’. The exception is the “General” format, which is localized.
- **All other Data types:** all other Data types map to the text formats for Double. The “Default” format uses a numeric format where the maximum number of significant digits comes from User Preferences (see section 1.8.4). For example, if the axis Data type is Current and the User Preferences is set to 3 digits for current, then the axis will display up to three digits of precision for the axis labels. This is only true for the “Default” format.
- **Draw\_ Options:** these checkboxes allow you to turn on or off various visual elements for the axis.

#### 1.6.4.2 Signal Analysis Sequence Step

The Signal Analysis sequence step processes time-based signal data from another measurement. The step performs three operations:

- **Smooth data:** applies a moving window average smoothing algorithm to the data (optional). If applied, the specified number of data points will be averaged together across the data set to reduce noise in the signal. Smoothed data will be used for all further processing.
- **Histogram settings:** calculates and displays a histogram of the data. The histogram sorts the signal data into buckets, then displays a column (bar) chart showing the count of data points in each bucket. The number of buckets is either automatically selected or can be manually set. For each bucket, the histogram calculates the range of the bucket (minimum, maximum, and median), and the count of data points in the bucket.
- **Fast Fourier transform (FFT):** performs an FFT on the data and display the results. The resulting data is a table containing frequency, spectrum magnitude, and power spectral density.



**The FFT operation depends on having an accurate time base for the signal. Some instruments, like the Lake Shore M81-SSM Synchronous Source Measure System, can collect accurate time data as part of the measurement.**

Each of these operations generates a new measurement results table that can be used for further operations (in a Chart step for instance), or exported for further analysis.



This step cannot be performed inside a loop, and can only be added and positioned at the step at the top level of the sequence.

#### 1.6.4.2.1 Signal Analysis settings

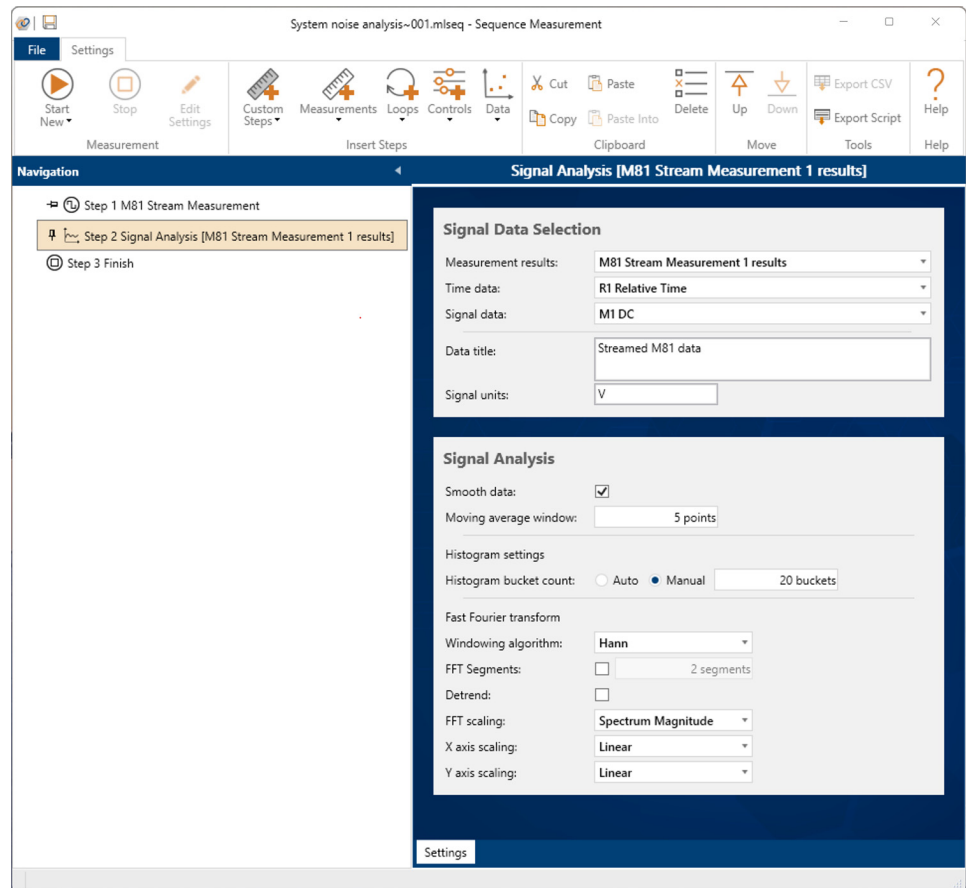


FIGURE 1-49 Signal Analysis settings

The settings are divided into two sections, as listed below.

- **Signal Data Selection:** choose the data you wish to analyze.
  - *Measurement results:* select the measurement data you wish to process. This must be the results from another measurement in the sequence. The drop down list shows all possible results tables.

**Analysis can only be done after the data is collected. Therefore, the Signal Analysis step needs to be located after the source measurement in the sequence. A validation error will appear if this step is moved before the measurement.**



**At this time, data from inside a loop cannot be used for signal analysis.**

- *Time data:* select the column of data to use as a time base for the analysis. Any column whose data can translate into a number can be used, but if the data does not represent some sort of signal base, the results may not be as expected.
- *Signal data:* select the column of data containing the signal you wish to analyze. Any column whose data can translate into a number can be used.
- **Signal Analysis:** controls how the processing is completed.

- *Smooth data*: when checked, the source signal will have a moving window average smoothing algorithm applied. The smoothed data is *not* used for any analysis.
- *Moving average window*: sets the size of the smoothing algorithm window. For example, if the window size is three points, then the data will be smoothed by averaging three raw data points together.
- *Histogram bucket count*: controls how the histogram is calculated. If Auto is selected, the number of buckets is determined by taking the square root of the number of data points. If the result is less than 5, then there will be 5 buckets. If the result is more than 20, then there will be 20 buckets. If Manual is selected, the number of buckets must be set at least 2).
- *Windowing algorithm*: selects the windowing function to use during the FFT calculation. The following algorithms are supported:
  - None (uniform)
  - Blackman
  - Blackman-Harris
  - Blackman Exact
  - Flat Top
  - Hamming
  - Hann
- *FFT Segments*: for large data sets, it can be useful to divide the set into segments and perform the FFT on each segment, and then combine the results. If this option is selected, the data will be split into the specified number of segments.
- *Detrend*: if selected, each signal data point will have the average of all the signal data subtracted from the point before calculating the FFT. If FFT segments are used, the average will be of all the data, not each segment's average.
- *FFT scaling*: choose to either display Spectrum Magnitude or Power Spectral Density in the FFT chart.
- *X/Y axis scaling*: the FFT chart can have either a linear or logarithmic axis scaling.

**Output:** either two or three measurement results tables and three charts are generated.

- Tables:
  1. If Smooth data is selected, a table will be generated with the same name as the source data with "(smoothed)" after it. The data is the same as the source data except that the signal column will have the moving average smoothing applied.
  2. Histogram bucket information.
  3. FFT spectrum magnitude and power spectral density.
- Charts:
  1. Signal being analyzed. If smoothing is selected, it will show the smoothed data.
  2. Histogram column chart with signal amplitude on the X axis and bucket count on the Y axis.
  3. FFT spectrum magnitude vs. frequency or power spectral density vs. frequency.

## 1.7 Scripts

MeasureLINK™-MCS software hosts an integrated scripting language based on Microsoft® Visual Basic.NET. This gives you the ability to develop custom solutions to collect, process, and present data. Any measurement sequence can be exported to a script, modified, and executed. External instruments and tools can also be accessed from a script and integrated into measurements.

MeasureLINK™-MCS software can be downloaded from:

<https://www.lakeshore.com/software>. Many example scripts are included with the software. The API documentation contains working examples for the API calls. Other sample scripts can be found in the MeasureLINK folder on your hard drive: (C:\Users\Public\Documents\MeasureLINK\Examples). Scripts for any registered measurements (section 1.7.7) can be opened and examined or modified.

### 1.7.1 Script Measurement Concepts

Script files contain the script code, data output structures including charts, tables and reports, and any data collected by the script. Scripting is managed through the Script window and script files (\*.mlscr). MeasureLINK™ exposes functionality to the scripting language via an application programming interface or API. MeasureLINK™ provides an API to prompt the user for information and allow the collection, manipulation, and display of measurement data. Device drivers expose their own APIs to control measurement hardware. The script language can also use APIs from third-party software modules or assemblies provided by the user or a third-party script developer. For example, a curve fit could be performed by the open-source Math.Net Numerics library: <https://numerics.mathdotnet.com>.

The Script window is fully capable of running a measurement, but is really intended for developing and testing measurements. Once the script is complete, it can then be used as a building block in a Measurement Sequence without having to know the scripting language or the details of how the measurement works.

### 1.7.2 Script Window Features

The features of the Script window are shown and described below.

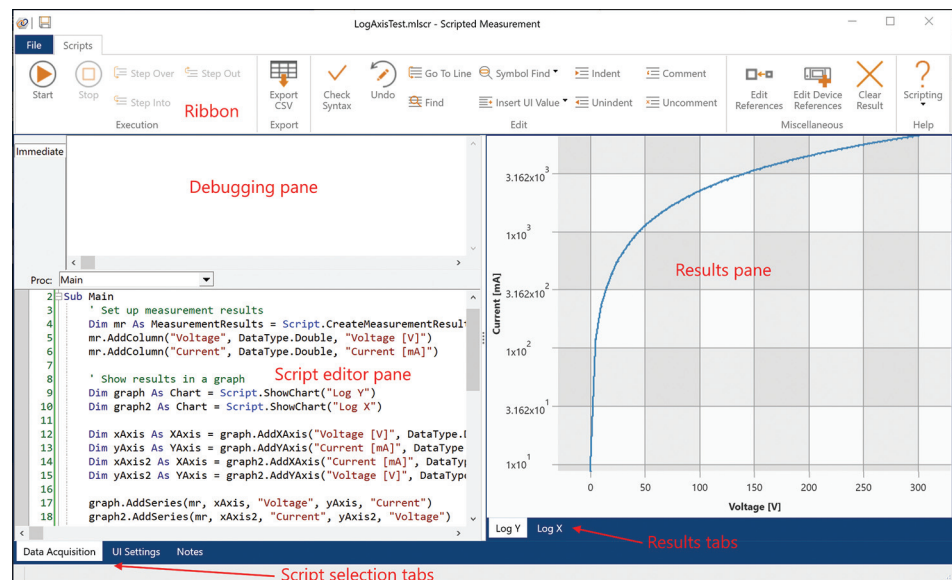


FIGURE 1-50 Scripting

The Script window is divided into six sections, as shown in the figure above:

- Script selection tabs, see section 1.7.2.1
- Results pane, see section 1.7.2.2
- Script editor pane, see section 1.7.2.3
- Script navigation controls, see section 1.7.2.4

- Debugging pane, see section 1.7.2.5
- Ribbon, see section 1.7.2.6



The sections of the Script window are divided by splitter controls. Click and drag the controls to adjust the relative sizes of the adjoining sections.



Press F1 in the Editor pane to open the language reference Help file.

### 1.7.2.1 Script Selection Tabs

Every script file contains two scripts:

- *Data acquisition script*: used to perform the measurement and present the data.
- *User interface (UI) script*: used to display controls so the user can modify measurement parameters.

The two scripts run independently of each other. Whichever tab is selected is the current active script that will run when you click the Start button. To select a script, click the corresponding tab. The API is slightly different for the data acquisition and UI scripts.

There is also a Notes tab which provides a place for keeping track of any additional information in a script file. You can add both formatted text and images.

### 1.7.2.2 Results Pane

This pane only appears if the script requests either a user interface (for a UI script) or one or more results tabs (for an acquisition script). A new script, or a script that does not show any UI or data, does not have a results pane. Acquisition results can include charts, tables, reports, and calculated values. The script developer can create any number of result tabs on the results pane and arrange the results on each tab as required. Results can be cleared at any time using the Clear Result button on the ribbon.

### 1.7.2.3 Script Editor Pane

Use the script editor pane to create or change the script logic.

```

1 Sub Main
2     Debug.Clear
3     FieldController.StartGoToField(Field.Tesla(0.52))
4
5     Do
6         Wait 5
7         Debug.WriteLine($"Field at {DateTime.Now} is {FieldController.Field.ToString}")
8     Loop While FieldController.IsFieldRamping
9
10    FieldController.StartGoToField(Field.Tesla(0.0))
11 End Sub

```

#### 1.7.2.3.1 Context coloring

Different elements of the code are automatically colored by the editor to help with readability. For example, comments are green and language keywords are in blue. In addition, a color bar on the left side of the pane shows yellow for lines of code that have changed since the script last ran or was saved.



### 1.7.2.3.2 Autocomplete

The editor provides an autocomplete context tool tip which appears whenever the editor recognizes what is typed. If multiple options exist, they all appear. In addition, changes to a line are automatically capitalized and highlighted when a different line is selected. Manually start autocomplete by pressing Ctrl plus the space bar in the Editor.

```

1 Sub Main
2     ' Define the first segment schema.
3     Dim segment1 As MeasurementResults = Script.CreateMeasurementResults("Segment 1")
4
5     Dim firstColumn As String = "Field"
6     Dim secondColumn As String = "Moment"
7     Dim thirdColumn As String = "Length"
8
9     '
10
11
12
13
14
15
16
17
18
19
20 Dim tab1 As ResultsTab = Script.CreateResultsTab("Big Tab", 2, 3)
21 tab1.Show()

```

### 1.7.2.3.3 Status and breakpoint bars

There are two gray bars, one on the left side of the editor pane and one below the editor pane. The one on the left is the breakpoint bar. It shows status information during debugging like the locations of breakpoints and the current line of execution. The status bar at the bottom shows messages from the editor during some operations. There is also a panel on the right side of the status bar that shows the line number you are currently on.

### 1.7.2.4 Script Navigation

The script navigation combo box navigates to methods in the script. They also reflect the current procedure (Proc) where the cursor or debugger is located.

```

Proc: Main
(declarations)
Main
1 Define the first segment schema.
2 Dim segment1 As MeasurementResults = Script.CreateMeasurementResults("Segment 1")
3
4
5 Dim firstColumn As String = "Field"
6 Dim secondColumn As String = "Moment"
7 Dim thirdColumn As String = "Length"
8 segment1.AddColumn(firstColumn, DataType.Double)
9 segment1.AddColumn(secondColumn, DataType.Double)
10 segment1.AddColumn(thirdColumn, DataType.Double)

```

### 1.7.2.5 Debugging Pane

The Script window includes a powerful integrated debugger to help test and develop scripts. The debugger pane contains several pages that are used with the debugger. See section 1.7.6 for more information.

### 1.7.2.6 Ribbon

The controls on the ribbon are used to operate the Script window:

Execution tools	<b>Start</b>	Works for both the data acquisition the UI script. This is only enabled if no other acquisition is running on the system and the software is not in viewer mode. The script can also be run by right-clicking in the Script window and clicking <b>Run</b> , or by pressing <b>F5</b> .  If the debugger is in use stepping through code, <b>Start</b> resumes execution of the script. The script runs until it completes or a breakpoint is hit.
	<b>Stop</b>	Stops the currently running script. <b>Stop</b> is enabled if any script is running. <b>Stop</b> immediately aborts any running script.
	<b>Step Over</b>	While debugging, steps over the current line. If the line is a call to a function or subroutine, <b>Step Over</b> executes that procedure, then advances the debugger to the line after the procedure. Otherwise, it advances the debugger to the next line. This can also be done by pressing <b>Shift + F8</b> .
	<b>Step Into</b>	While debugging, steps into the current line. If the line is a call to a function or subroutine, <b>Step Into</b> advances the debugger into the new procedure. Otherwise, it advances the debugger to the next line. This can also be done by pressing <b>F8</b> .
	<b>Step Out</b>	While debugging, advances the debugger until the current procedure is complete. This can also be done from the context menu.
Export tools	<b>Export CSV</b>	Exports results from the Script window to a comma separated values text file. If the script file contains multiple measurement results, you will be prompted to select the results to export.
Edit tools	<b>Check Syntax</b>	Scans the current script (data acquisition or UI setting) and checks for any syntax errors. This button is disabled while running or debugging a script. If syntax errors are found, a dialog appears listing the errors. If everything is correct, the message “No syntax errors” appears in the status bar of the editor pane.
	<b>Undo</b>	Reverts the last change(s) made in the script editor.
	<b>Go to Line</b>	Moves the cursor to the start of the specified line number in the code.
	<b>Find</b>	Searches the code for the next instance that matches the given text.
	<b>Symbol Find</b>	Allows navigation to references to symbols like variables, constants, and subroutines. Clicking a symbol in the editor highlights all references to that symbol. You can then use the menu items to navigate to symbol references.
	<b>Insert UI Value</b>	Displays a searchable list of UI values defined in the UI Script [insert link/reference here]. Clicking the green + icon automatically inserts a reference to that UI value in the script at the current cursor location.
	<b>Indent</b>	Indents the selected lines of code, four spaces to the right.
	<b>Unindent</b>	Unindents the selected lines of code by moving them four spaces to the left.
	<b>Comment</b>	Comments out the currently selected lines of code. This is useful for temporarily preventing them from executing.
	<b>Uncomment</b>	Uncomments the currently selected lines of code.
Miscellaneous tools	<b>Edit References</b>	Opens the References dialog to show the current script's referenced .NET assemblies and COM unnecessary libraries. Several of the software's components are referenced by default. Additional references can be added to external instrument drivers, numeric libraries, or any other third-party or custom component you wish to access from the script.
	<b>Edit Device References</b>	Opens the Script Device References to allow you to reference a hardware device from the script. Adding a device reference allows you to use the device from script but also requires that device to be present and connected before the script will run. See <a href="#">section 1.7.3</a> below for details
	<b>Clear Result</b>	Clear any measurement results, UI values, and result tabs from the current script file. <b>CAUTION:</b> The cleared data is permanently deleted.

TABLE 1-1 *Script ribbon*

## 1.7.3 Referencing Hardware



Most measurements require interacting with some sort of instrumentation. Use the **Edit Device References** button on the ribbon to choose which devices you are using so that the API of those instruments will be available to the script.

**The devices and drivers available depend on your active system configuration (see section 1.8.3), and your installed application packs (see section 1.9.1).**

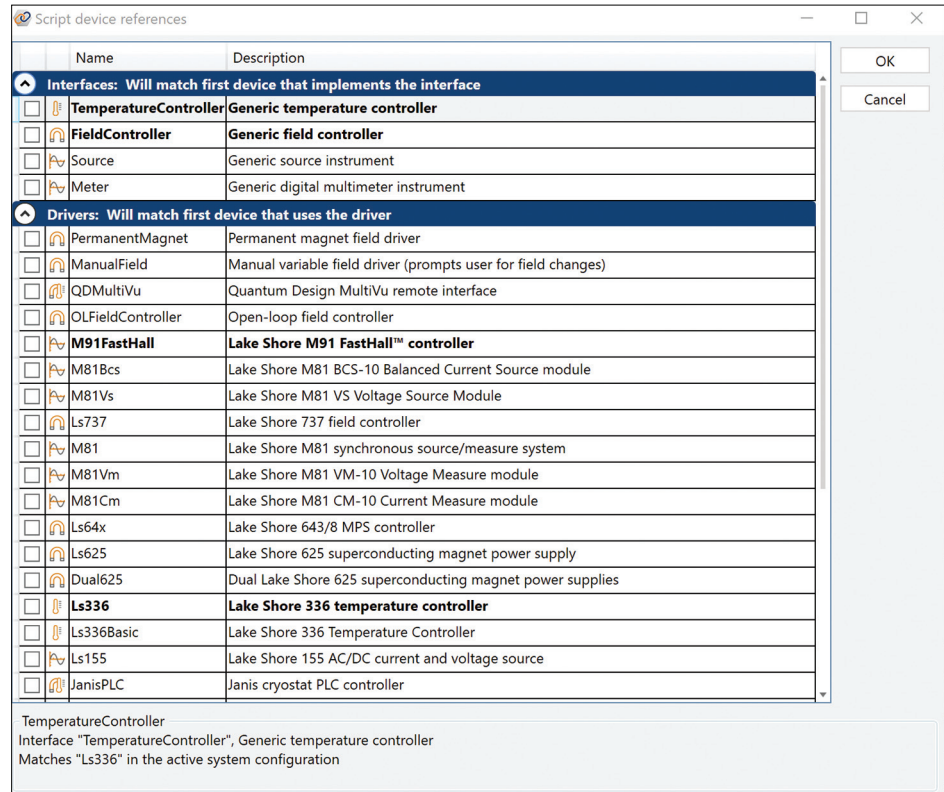


FIGURE 1-51 Script devices references

The Script device references dialog box lists three categories of device references: Interface, Driver, and Device. Choose a category based on your script's requirements. Each category is explained below. To include a device reference, check the checkbox for that device reference.

- Interface:** use this option if the script works on any device whose driver implements the selected interface. An interface is the most general but least full-featured category of any device that supports that interface. Interfaces provide very generic, high-level operations, which makes them more flexible. If you write your script against the interface when the script is run, the software will find the first connected device that supports that interface and use it. For example, a script that references the TemperatureController interface could control temperature on any temperature controller whose driver implements the TemperatureController interface.
- Driver:** use this option if the script works only on a specific driver, such as the Ls336 driver written to communicate with a Lake Shore Model 336 temperature controller. The driver category is less general than an interface but still allows for flexibility. All drivers installed in the MeasureLINK™-MCS software appear in this list. If you select a driver from this list, your script will have full access to all driver functionality. When the script is run, the software will search for the first configured device that uses the selected driver and control it.
- Device:** use this option if the script works with a specific named device on your system. The device category shows only devices currently in the active system configuration. Your script will only run if a device of the same type and name is

currently connected. This is the most specific category. This option must be used when communicating with multiple devices that use the same driver.

Remove device references by unchecking the box. To prevent the references from conflicting, the software disables options that conflict with your current selection. For example, if you add a reference to the TemperatureController reference, all drivers and devices that represent a TemperatureController are disabled. Likewise if you select a Model 336 driver, the TemperatureController checkbox and any named Model 336 devices will be disabled.

### 1.7.4 User Interface Scripts

User interface (UI) scripts can be used to collect information from the user before starting a measurement. To run a UI script, select the **UI Settings** tab and click the **Start** button. The user interface will appear in the results pane.

The simple UI script shown below collects two numeric values and a Boolean flag. The data collected will be stored as a UI value in the script file where it can be retrieved and used by the data acquisition script.

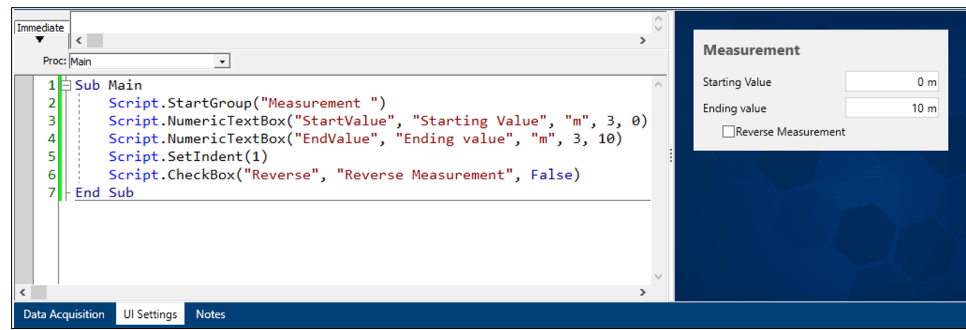


FIGURE 1-52 UI Settings script

Each control in the UI script has a control name that is used to store and retrieve the data entered in the control. From the acquisition script, click the **Insert UI Value** button on the ribbon to search UI values, and to insert codes to retrieve the control value into the script.

### 1.7.5 Data Acquisition Scripts

The data acquisition script below gets the data from the example UI and prints it in the debugging pane (see section 1.7.2.5).

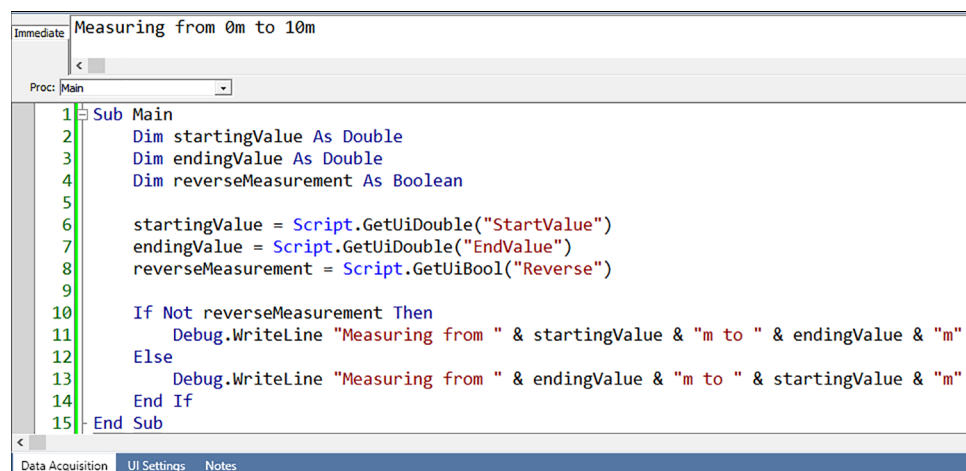


FIGURE 1-53 Data Acquisition script

In a data acquisition script, use data from the UI to setup your instruments, capture the data, and display the results. For example, this script will ramp a temperature controller to a temperature supplied by the user interface script UI values:

```
Sub Main
    Dim target As Double
    target = Script.GetUiDouble("TargetTemperature")
    TemperatureController.GoToTemperature(Temperature.Kelvin(target))
End Sub
```

Refer to the sample scripts for examples of how to do various operations in an acquisition script. Sample scripts can be found in the MeasureLINK folder on your hard drive: (C:\Users\Public\Documents\MeasureLINK\Examples).

## 1.7.6 Debugging a Script

Much of the functionality of the editor pane revolves around debugging.

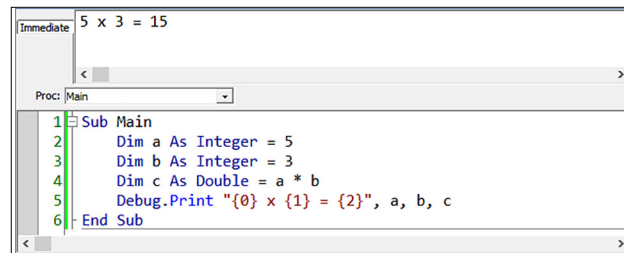
### 1.7.6.1 Debug.Print

To debug a script, insert "Debug.Print" into the code to show the value of the variables you want to see:

1. Open a new Script window.
2. Type the following code at line 2 in the **Data Acquisition** tab:

```
Dim a As Integer = 5
Dim b As Integer = 3
Dim c As Double = a * b
Debug.Print "{0} x {1} = {2}", a, b, c
```

3. The code should appear as follows:



4. Click **Check Syntax** on the ribbon to make sure there are no syntax errors.
5. Click **Start**.
6. The script runs and the **Immediate** tab updates.
7. The curly brackets in the "{0} x {1} = {2}" string evaluate the variables that follow it in the specified order. For example, {0} points to the first one (a) and {2} points to the last one (c).

### 1.7.6.2 Breakpoints

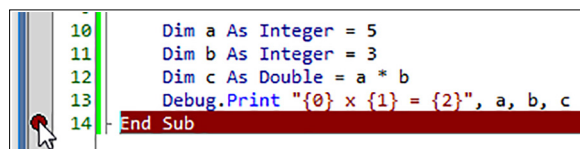
The software allows the user to set breakpoints in the script to stop and inspect the process. The debugger stops when it hits the first breakpoint. You can inspect variables while script execution pauses.



**A breakpoint does not work on an empty line.**

As an example, to add a breakpoint to the script:

1. Click line 14 in the gray, vertical bar:



2. The entire line is highlighted in red. This indicates that the script execution “breaks” when it encounters line 14.
3. Click **Start**.
4. The script execution stops at line 14 and the highlight changes to yellow:

```

10 Dim a As Integer = 5
11 Dim b As Integer = 3
12 Dim c As Double = a * b
13 Debug.Print "{0} x {1} = {2}", a, b, c
14 End Sub

```

5. The “Debug.Print” statement shows the result in the **Immediate** tab.
6. Type ?a and press the **Enter** key in the debug pane. This only works when the program is paused at a breakpoint.
7. The software shows “5%”, which means that the variable “a” is evaluated as an integer 5.
8. Type ?c and press the **Enter** key.
9. The software shows “15#”, which means that the variable “c” is evaluated as a double 15.
10. Click **Continue** on the ribbon.
11. If the file has not already been saved, the script execution finishes and a File Save dialog box appears.
12. Save the file. The file extension must be “.mlscr” for the software to recognize it as a script file. When you open the same file and run it again as is or after some modification, the file is automatically saved with the same file name when the script execution finishes.

### 1.7.6.3 Stopping the Script Execution

To stop the script execution before it finishes, click **Stop** on the ribbon.

### 1.7.6.4 Stepping through the Code

Once the debugger is stopped at a breakpoint, there are a few ways to continue execution after a breakpoint:

- **Continue** button: resumes the execution and may run to completion or break again when the script hits another breakpoint.
- **Step Over** button: single-steps the execution. The execution stops at the next executable line, even if the statement at the current breakpoint involves another subroutine or function call.
- **Step Into** button: single-steps the execution. If the statement at the current breakpoint involves another subroutine or function call, the execution stops at the beginning of the called subroutine or function.
- **Step Out** button: when the current breakpoint is inside a subroutine or function, the execution stops at the next executable line after the line where the caller calls the subroutine or function.

### 1.7.6.5 Using the Debugging Pane Tabs

While debugging, examine and change the state of the running script using the tabs on the debugging pane. The three tabs are described below:

#### 1.7.6.5.1 Immediate tab

The **Immediate** tab is the only debugging pane that is available all the time. It is used for:

- **Displaying data:** scripts can display any desired text in the **Immediate** tab using the **Debug** object. This is listed in several places in the example script:

```
Debug.Print "This is a script"
```

Running the entire example adds three status lines in the **Immediate** tab. The **Immediate** window is useful for displaying status messages, calculated values, or

intermediate results from inside a script. The debug object includes the following methods:

Method	Description
<b>Clear</b>	Clears the <b>immediate</b> tab of all output.
<b>Print</b>	Displays data with formatting capability; advances to the next line.
<b>Write</b>	Displays data but does not advance to the next line.
<b>WriteLine</b>	Displays data and advances to the next time.

- *Evaluating an expression:* any script expression can be evaluated and the result displayed by typing “?” in the Immediate window followed by an expression. Some examples are listed below:

Expression	Result
?”Hello”	”Hello”
?s + 2	4%
?UCase(”xyz”)	”XYZ”
?4 = 5	False

- *Assigning a variable:* variables can be assigned in the Immediate tab by typing “**var = expression**” to assign a variable by value, or “**Set var = expression**” to assign a value by reference. This only works when the program is stopped at a breakpoint.
- *Calling a procedure:* call any procedure including the built-in ones by simply typing the call into the Immediate tab. This only works when the program is stopped at a breakpoint.

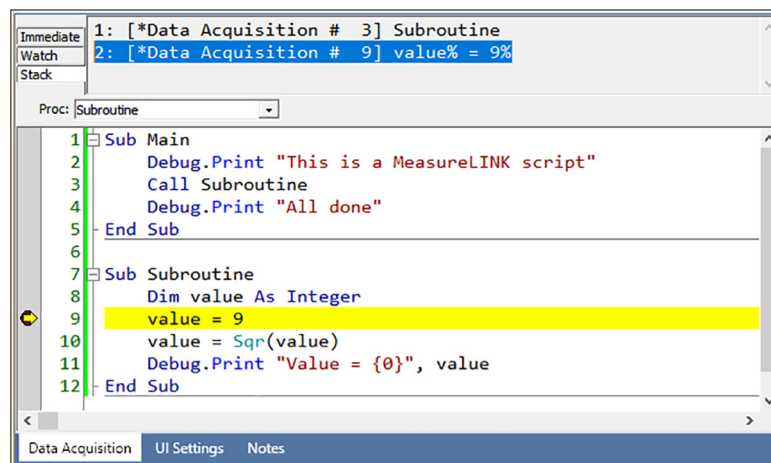
#### 1.7.6.5.2 Watch tab

The **Watch** tab only appears while debugging. This tab allows you to monitor the values of variables as you step through the script. To add a watch, right-click a variable and select **Add watch**. The variable and its current value appears in the Watch window.

If the variable changes, the **Watch** tab value updates to reflect the new value. To stop watching a variable, select the watch text in the **Watch** tab and delete it. Press **Enter** in the Watch window to force all values to update immediately. Change the value of the variable by changing the value displayed in the **Watch** tab.

#### 1.7.6.5.3 Stack tab

The **Stack** tab shows the call stack for the current statement. In the example, if the debugger is stopped in the subroutine, the call stack looks as follows:



## 1.7.7 Registering Script Files

Any script file can be registered for use as a measurement sequence step. Registered scripts are managed from MeasureLINK™-MCS software. To get to the registration utility, go to the navigation pane and select Measurement Registration. Registered script files can be per user, shared with all users, or pre-installed with MeasureLINK™. Once a script is registered, it will appear in the drop down list under Measurement on the ribbon for the Sequence Measurement window.

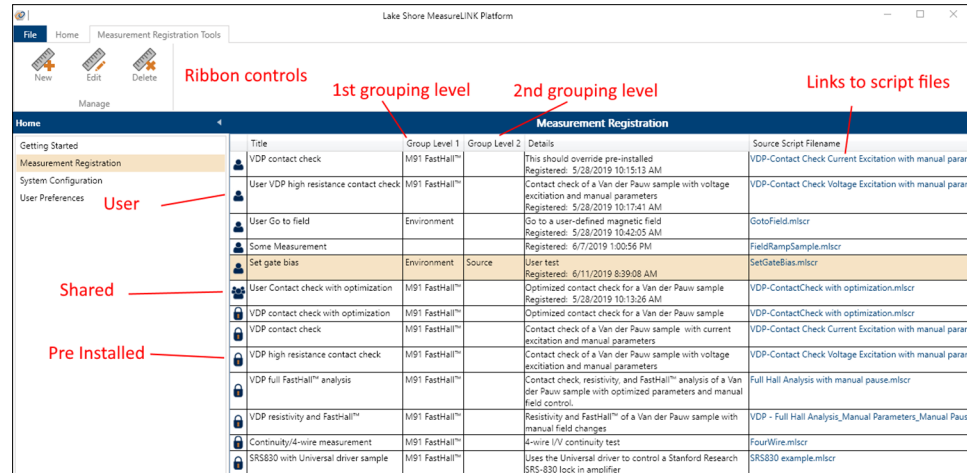


FIGURE 1-54 Registering script files

### 1.7.7.1 Registering a Script

1. From the Measurement Registration utility, click **New** on the ribbon. The Custom Measurement dialog box will appear.
2. Select the script file you wish to add and provide a title and (optional) a description for the new custom measurement step.

Register Script as Measurement

Script file name: C:\Users\nathan.vonada\Documents\MeasureLINK\SetGateBias.miscr [Browse...]

Measurement title: Set gate bias

Description: User test

Grouping

Select or enter the group levels under which to display the script in the Measurement menu on the Sequence Measurement toolbar.

Level 1: Environment

Level 2: Source

Sharing

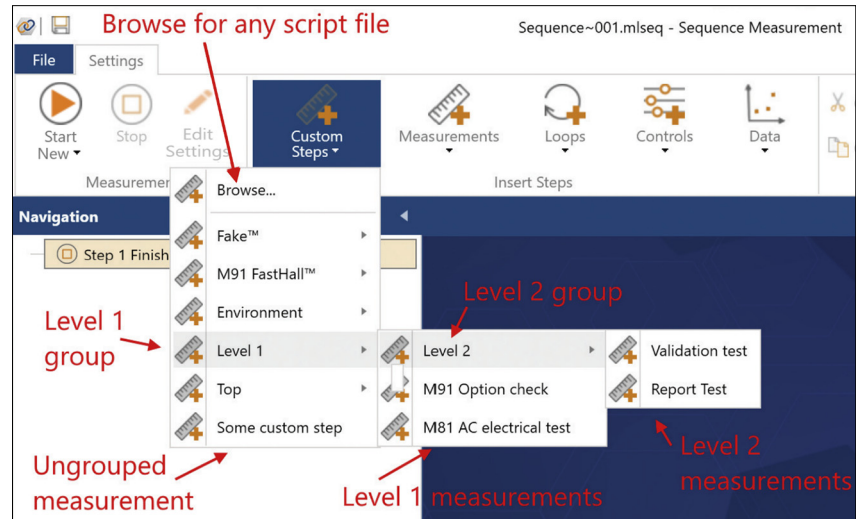
If you want to share a registration with all users be sure other users have read access permissions for the script file.

Share registered script with all users

OK Cancel

3. Set grouping levels (optional) for the registered measurement. Two grouping levels are supported. Groups appear as sub-menus under Measurement on the ribbon (see the image below).





4. You can also specify the type of registration. There are three types of registered measurements:
  - **User registration:** this registration will only appear for the current Windows® user.
  - **Shared registration:** appears for all Windows® users. The script file should be stored in a location all users can access (such as the Public Documents\MeasureLINK folder).
  - **Pre-installed registrations:** measurement scripts installed by Lake Shore with the product or a MeasureLINK™ application pack (see section 1.9.1). They are read-only and cannot be created, modified or deleted. However, it is possible to supplant a pre-installed script, listed under **Measurement** on the ribbon, by creating a shared registration or user registration that has the same title and grouping levels as the pre-installed script.
5. Click OK. The new measurement will appear in the list of registered scripts.



The registered script is not copied, moved or changed in any way. If you modify the script, the modified script will be used in sequences. If you delete or move the script file, the registration will be broken.

#### 1.7.7.2 Editing a Script Registration

Changing a script registration does not change existing sequences or the source script file in any way. To edit a script registration:

1. From the Measurement Registration window, select the registration you wish to edit in the list of registered scripts.
2. Click **Edit** on the ribbon.
3. A dialog box appears. Modify any information you wish to change.
4. Click **OK** to save the updated registration.

#### 1.7.7.3 Deleting a Script Registration

Deleting a script registration does not change existing sequences or delete the source script file. To delete a script registration:

1. From the Measurement Registration window, select the registration you wish to delete in the list of registered scripts
2. Click **Delete** on the ribbon or press the **Delete** key. A dialog box appears, confirming the deletion.
3. Click **OK** to delete the selected registration.

## 1.8 Home Level

The Home level includes:

- Getting Started
- Measurement Registration
- System Configuration
- User Preferences

### 1.8.1 Getting Started

The Getting Started pane includes links to documentation that may be useful in using MeasureLINK™. Also listed are recent files. Click a file name to open the file for use. These documentation links and recent files can also be found in the Backstage View (see section 1.3.6).

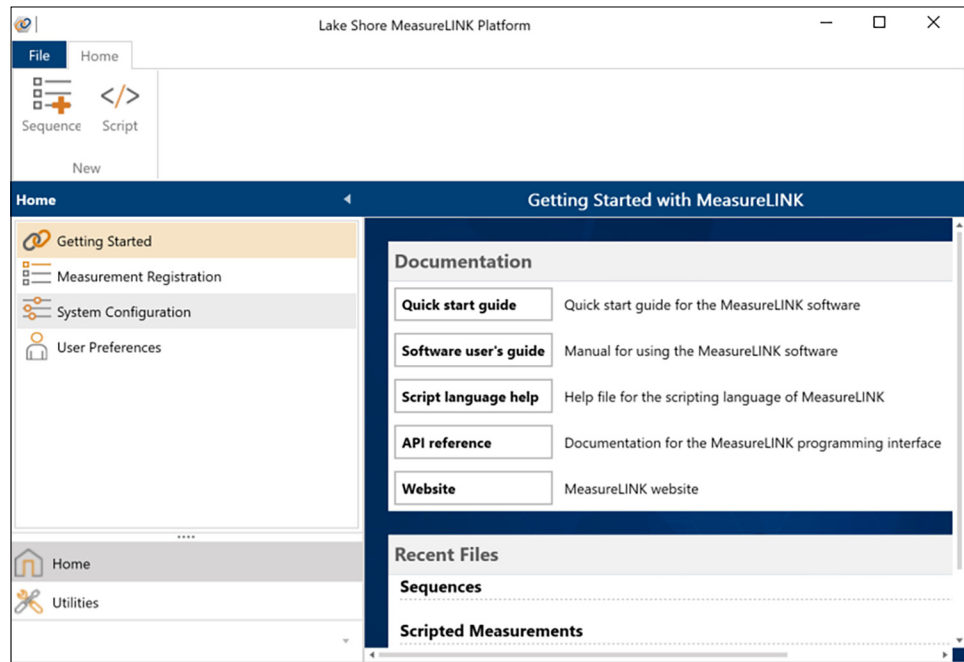


FIGURE 1-55 Getting Started

### 1.8.2 Measurement Registration

Files can be registered with the system for use as a measurement sequence step. Registered scripts are managed from MeasureLINK™-MCS software. To get to the registration utility, go to the navigation pane and select Measurement Registration. Once a custom script is registered, it will appear in the drop down list under Measurement Registration in the navigation pane. See section 1.7.7 for a detailed explanation of registered scripts.

### 1.8.3 System Configuration

The MeasureLINK™-MCS software allows the user to have multiple system configurations. A system configuration is a selection of instruments and the settings for those instruments. The controls on the System Configuration window allow you to create and modify configurations.

- **System Configuration ribbon:** contains options for controlling configurations.
- **Workspace:** the right side of the screen contains a tab control where the software will display settings.

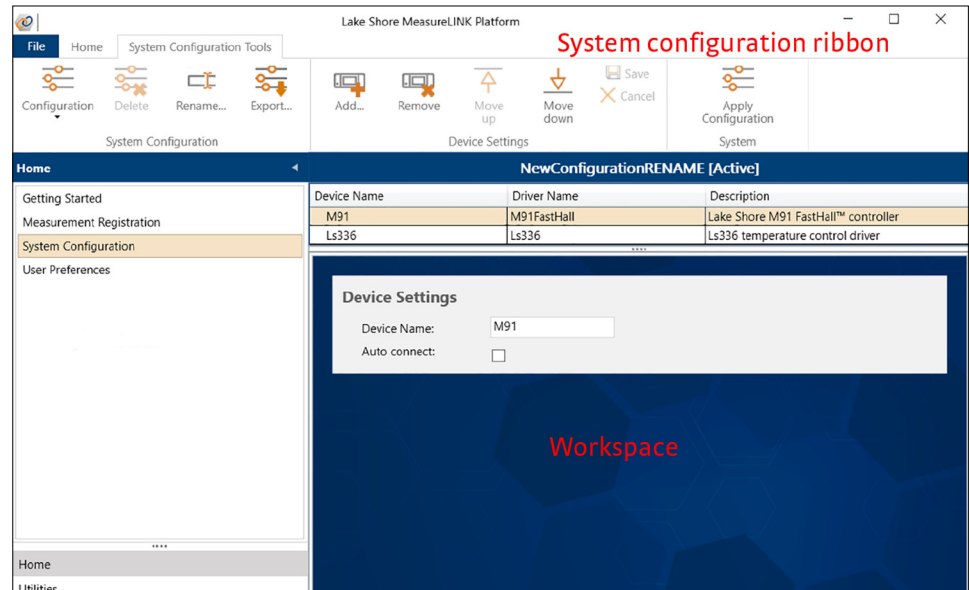


FIGURE 1-56 System Configuration window

#### 1.8.3.1 Configuration Controls

Click the Configuration button in the System Configuration ribbon to see the options.

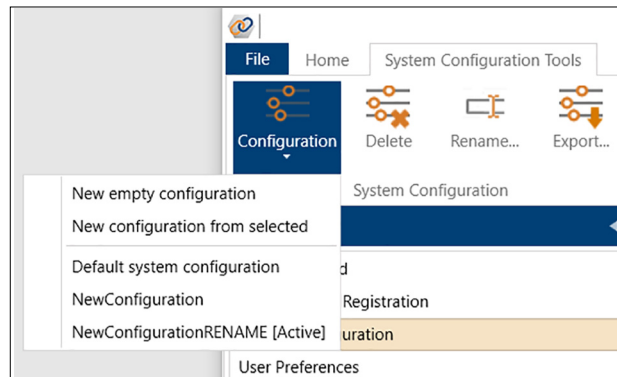


FIGURE 1-57 Configuration menu

The current configurations are shown, with the in-use configuration marked as [Active].

##### 1.8.3.1.1 Creating a New Empty Configuration

To create a new configuration:

1. Click the Configuration button.
2. Click **New empty configuration** from the drop down list.
3. The NewConfiguration window appears.
4. Modify any settings you wish to change. Click **Save** to save the configuration, or click **Apply Configuration** to apply the current configuration to the system.

### 1.8.3.1.2 Creating a New Configuration from a Selected Configuration

Use this option to copy a current configuration and save it as a new configuration:

1. Click the **Configuration** button.
2. Click **New configuration from selected** from the drop down list.
3. The NewConfiguration window appears.
4. Modify any settings you wish to change. Click **Save** to save the configuration, or click **Apply Configuration** to apply the current configuration to the system.

### 1.8.3.1.3 Deleting a Configuration

Use the **Delete** button to delete the currently selected, non-active, system configuration from MeasureLINK™-MCS software. To delete a configuration:

1. Open the configuration you wish to delete.
2. Click the **Delete** button.
3. A dialog box appears, confirming the deletion.
4. Click **OK**.
5. The configuration is deleted.

### 1.8.3.1.4 Renaming a Configuration

Use the **Rename** button to rename the currently selected system configuration:

1. Open the configuration you wish to delete.
2. Click the **Rename** button.
3. A dialog box appears.

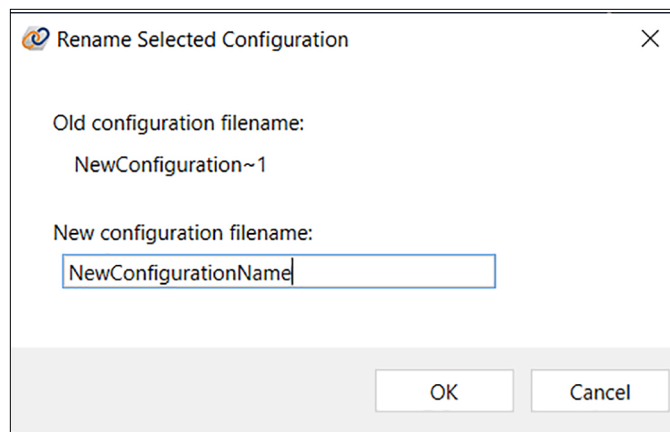


FIGURE 1-58 *Rename button*

4. Choose a filename and then click **OK**.
5. The configuration is renamed.

### 1.8.3.1.5 Exporting a Configuration

Use the **Export** button to export the currently selected system configuration as a JSON file, which can be shared and imported. To export a configuration:

1. Open the configuration you wish to export.
2. Click the **Export** button.
3. A Save As dialog box appears. Use this name or change it to another name if you prefer.
4. Choose the desired file location and name, then click **Save**.

### 1.8.3.1.6 Importing a Configuration

Use the **Import** button to import a system configuration from a valid JSON file. To import a configuration:

1. Click the **Import** button.
2. A Open File dialog box appears. Navigate to the JSON file you wish to import.

3. Select the JSON file and click **Open**.
4. If there is already a system configuration with the same name as the one being imported, a new name will be assigned to the imported configuration. Click **OK**.



The imported configuration will be selected, but not applied as the current configuration.

### 1.8.3.2 Device Settings

The Device Settings group on the ribbon allows you to control the settings in a configuration.

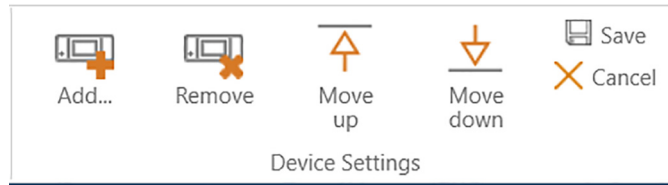


FIGURE 1-59 Device settings

#### 1.8.3.2.1 Add

To add a device to the configuration:

1. Click the **Add** button.
2. The Add Driver(s) pop-up window appears.

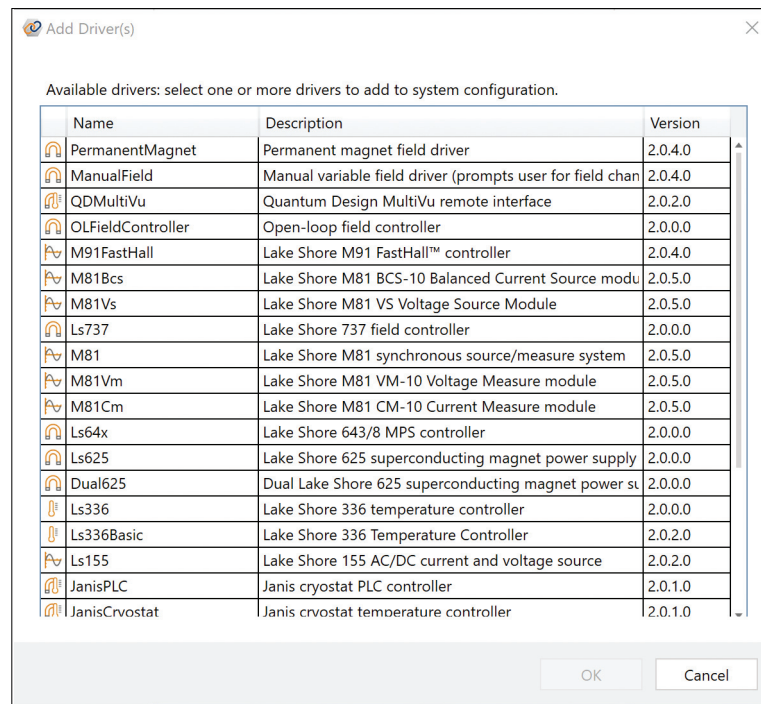


FIGURE 1-60 Add Driver(s)

3. Click a driver name to add it.
4. Click **OK**.
5. The device is added to the configuration.

#### 1.8.3.2.2 Remove

Use the **Remove** button to remove the currently selected device from the system configuration:

1. Select the device name that you wish to remove.
2. Click the **Remove** button.
3. A dialog box appears, confirming the deletion.

4. Click OK.
5. The device is removed.

#### 1.8.3.2.3 Moving Up or Down

Use the Move up and Move down buttons to shift the currently selected device up or down in the list of devices. The order in which the devices are displayed is the order in which they will be displayed in the Monitor window.

#### 1.8.3.2.4 Save

Click the Save button to save the currently selected system configuration.

#### 1.8.3.2.5 Cancel

Use the Cancel button to discard any changes made to the currently selected system configuration, and return to the last saved configuration:

1. Click the Cancel button.
2. A dialog box appears, confirming the action.
3. Click Yes.
4. The change is discarded.

### 1.8.3.3 System Tool Bar Settings

The System section of the ribbon allows you to apply the current configuration as the active configuration.

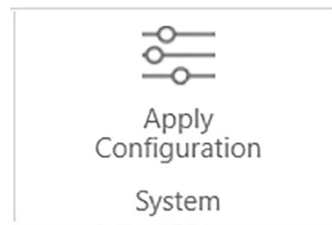


FIGURE 1-61 System tool bar

#### 1.8.3.3.1 Apply System Configuration

To apply a system configuration, click the **Apply Configuration** button. The software will either re-apply the active system configuration or apply the selected configuration, depending on the configuration that is currently selected.

#### Re-apply the active system configuration

The active configuration is the configuration that is currently applied to the system, as indicated by the “[Active]” tag appended to the configuration name in the title bar.

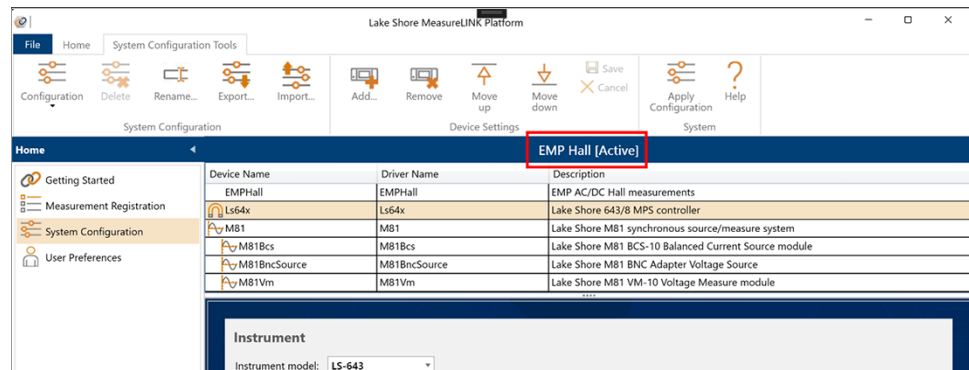
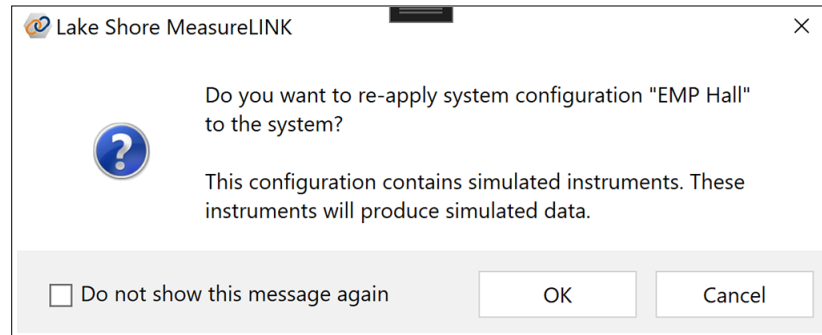


FIGURE 1-62 Active configuration

When this configuration is modified, it will need to be re-applied to the system in order for the modification to take effect. Click the **Apply Configuration** button while the configuration is selected, and the following confirmation dialog box will be displayed:



### NOTE

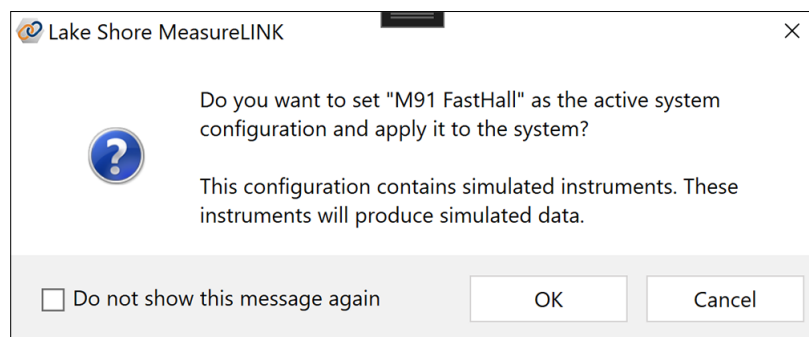
**If the active configuration contains devices that are configured to use simulation mode, the dialog box will contain a reminder message that data returned from these devices is not real but simulated.**

The dialog box may be disabled, if desired. Click the **Do not show this message again** checkbox before clicking the **OK** button, and the dialog will not be displayed in the future. This option can be reset from the User Preferences pane.

If the configuration has been edited when the button is clicked, the configuration will automatically be saved before it is applied. If the configuration has been edited and the **Save** button is clicked, the configuration will be saved, and then a dialog box offering to re-apply the configuration (as described above) will be displayed.

### Apply a selected configuration and set it as the active configuration

If a system configuration other than the active configuration is selected for editing using the **Configuration** button, the selected configuration can be applied to the system by clicking the **Apply Configuration** button. The following confirmation dialog box will be displayed:



### NOTE

**If the selected configuration contains devices that are configured to use simulation mode, the dialog box will contain a reminder message that data returned from these devices is not real but simulated.**

The dialog box may be disabled, if desired. Click the **Do not show this message again** checkbox before clicking the **OK** button, and the dialog will not be displayed in the future. This option can be reset from the User Preferences pane. If the configuration has been edited when the button is clicked, the configuration will automatically be saved before it is applied.

### 1.8.3.3.2 Devices List

Lists the configured devices (instrument drivers) for the configuration. Select a device to show its settings. The device tabs appear at the bottom of the screen. If Auto connect is not checked, the instrument will have to be manually connected from the monitor window.

The settings tabs that appear depend on the driver being configured. However, all drivers share the same first two standard tabs. These are explained below.

#### General tab

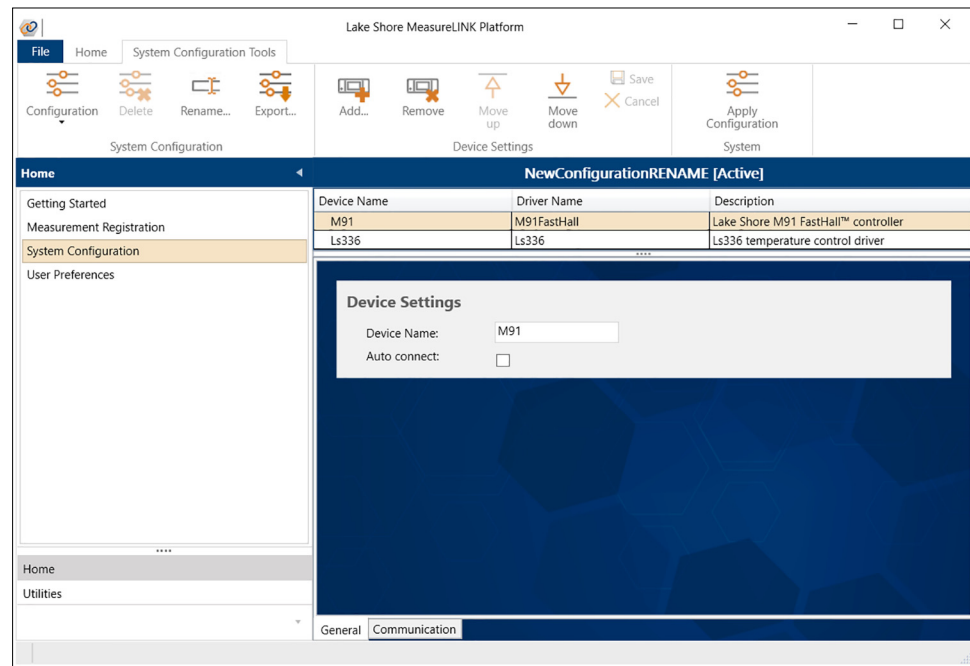


FIGURE 1-63 General tab

The following fields are available on the General tab:

- **Device Name:** this is the name by which the device is referenced from a measurement script. This name will appear in the Device section of the script window's device references dialog box (see section 1.7.3).
- **Auto Connect:** select this option to automatically connect the device upon software startup or when the configuration is applied.



## Communication tab

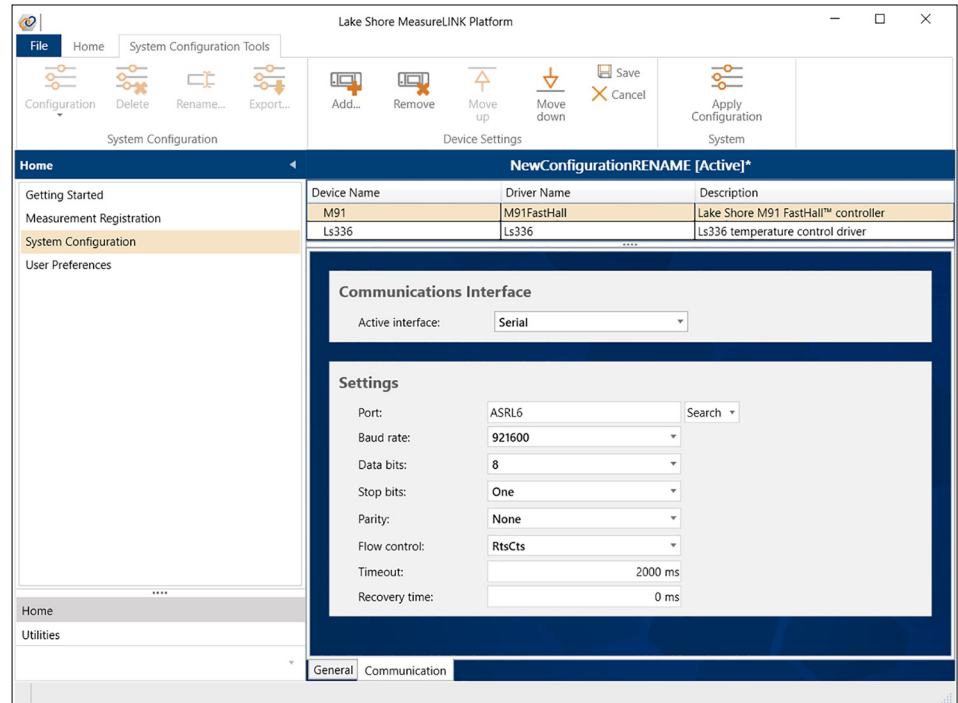



FIGURE 1-64 Communication tab

Select the communications interface to be used; the available interfaces are determined by the device driver. Then complete the settings for the chosen interface. For each field, choose a setting from the drop down list, or type a value in the text box.

### NOTE

Refer to the specific instrument user's manual to determine proper settings.

- **Simulator settings:** settings related to simulating the current instrument. Every instrument has a simulator that provides at least a basic simulation of how the instrument responds. Simulated instruments appear in the monitor window with an S icon 
- **Serial Settings:** settings related to communication through a serial port (COM or USB virtual serial port):
  - Port
  - Baud rate
  - Data bits
  - Stop bits
  - Parity
  - Flow control
  - Timeout
  - Recovery time
- **GPIB Settings:** settings related to communication through a GPIB interface:
  - Board
  - Address
  - Secondary address
  - Timeout
- **TCP/IP Settings:** settings related to communication through an IP address or host-name:
  - Board
  - IP Address
  - Port

- Keep alive
- No delay
- Timeout



**Not all settings can be changed for every instrument driver. The driver determines which settings are available and editable.**

### Test Device Communicator

When configuring the device communication settings for each of the devices in a system configuration, the settings can be tested using test device communicators. Use the **Test Connection** button on each interface settings page.

Device Name	Driver Name	Description
M81	M81	Lake Shore M81 synchronous source/measure system
M81Bcs	M81Bcs	Lake Shore M81 BCS-10 Balanced Current Source module
M81Vm	M81Vm	Lake Shore M81 VM-10 Voltage Measure module
F71	F71Teslameter	Lake Shore F71 or F41 teslameter

**Communications Interface**

Active interface: Serial (VISA) ▾

---

**Settings**

Port:  Search ▾

Baud rate: 921600 ▾

Data bits: 8 ▾

Stop bits: One ▾

Parity: None ▾

Flow control: RtsCts ▾

Timeout:  ms

Recovery time:  ms

Test Connection...

FIGURE 1-65 Test Connection button

Clicking this button brings up a device communicator dialog box, pre-configured to use the communication interface settings defined in the settings view from which it was launched.



**If the settings view corresponds to the active system configuration, and the selected device is already connected from the monitor pane, the button will be disabled in order to prevent any attempt to launch another connection to the same device.**

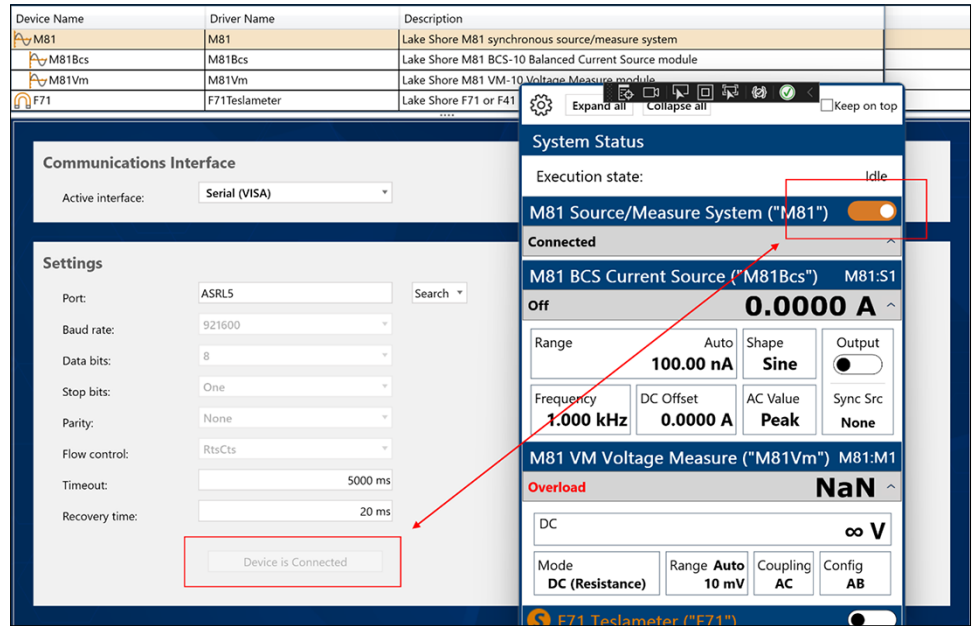


FIGURE 1-66 Test Connection button in a disabled state

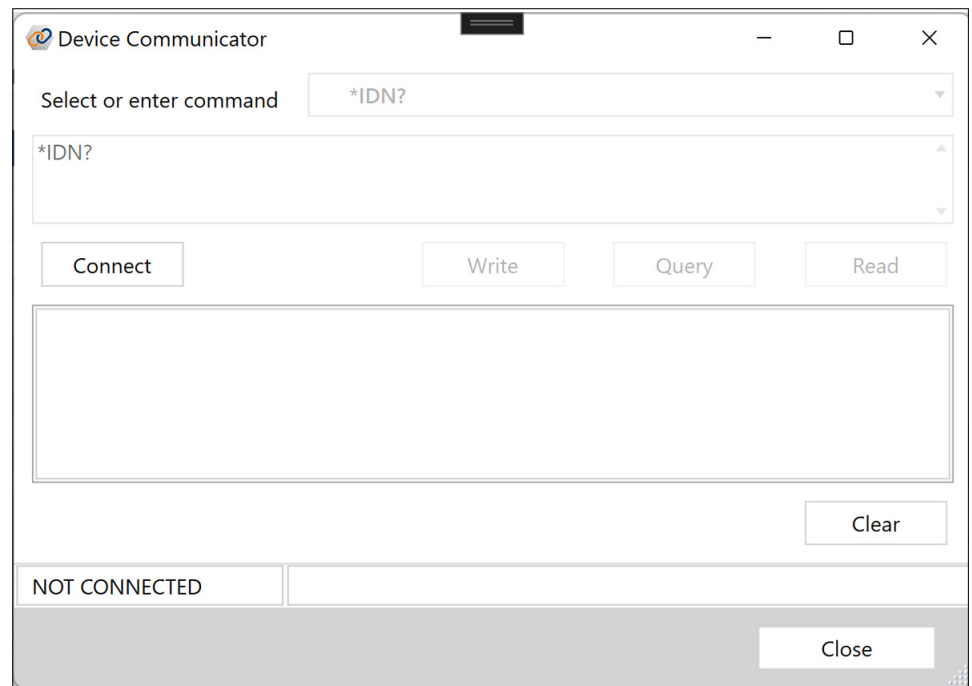


FIGURE 1-67 Device Communicator (disconnected)

When the Device Communicator is initially displayed, it is in a disconnected state. Press the **Connect** button to connect using the Communication Interface settings from the Settings view. If a connection is successfully completed, the dialog will enter a connected state. The Device Communicator will remain connected to the device until the dialog is closed, either by clicking the **Close** button, or the “X” in the upper right corner of the dialog.

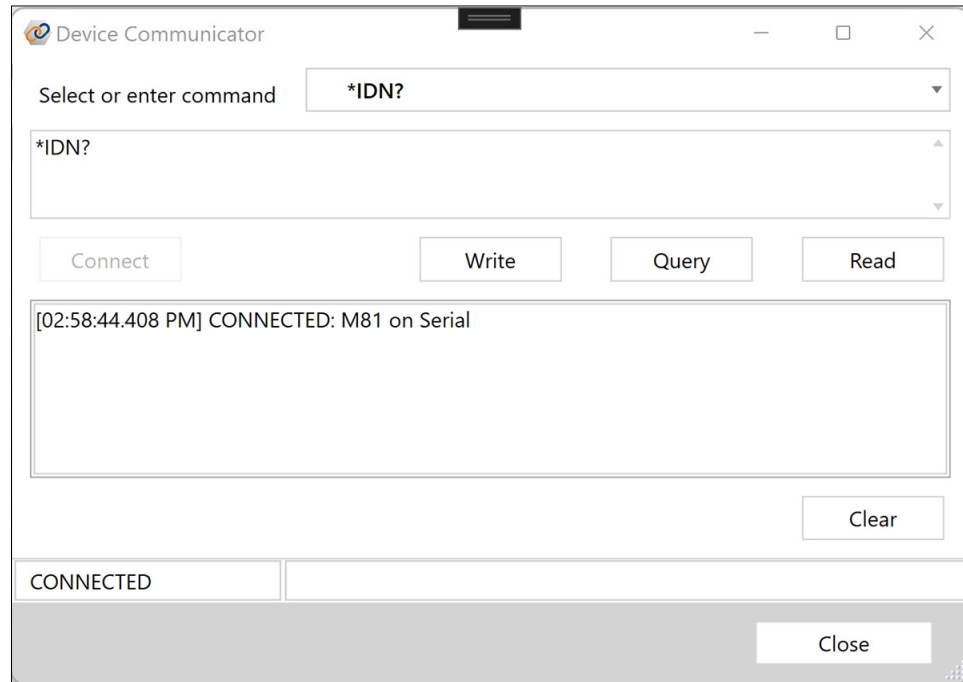


FIGURE 1-68 Device Communicator (connected)

If an error occurs while attempting to connect, the dialog box will remain in an unconnected state, and an error message will be displayed. Connection errors may result from incorrect communication interface settings, or from an attempt to connect to a device that is already in a connected state. When this occurs, the dialog should be closed, and the cause of the error should be resolved. Once resolved, the dialog can be reopened and another attempt to connect can be made.

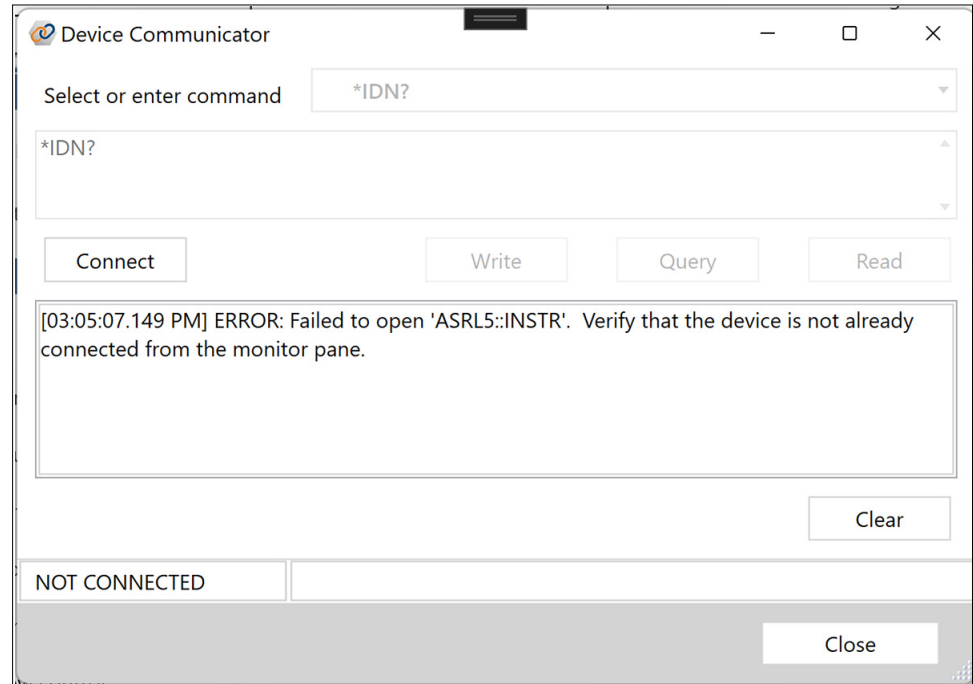


FIGURE 1-69 Device Communicator (connection error)

The device communicator dialog makes use of the following controls:

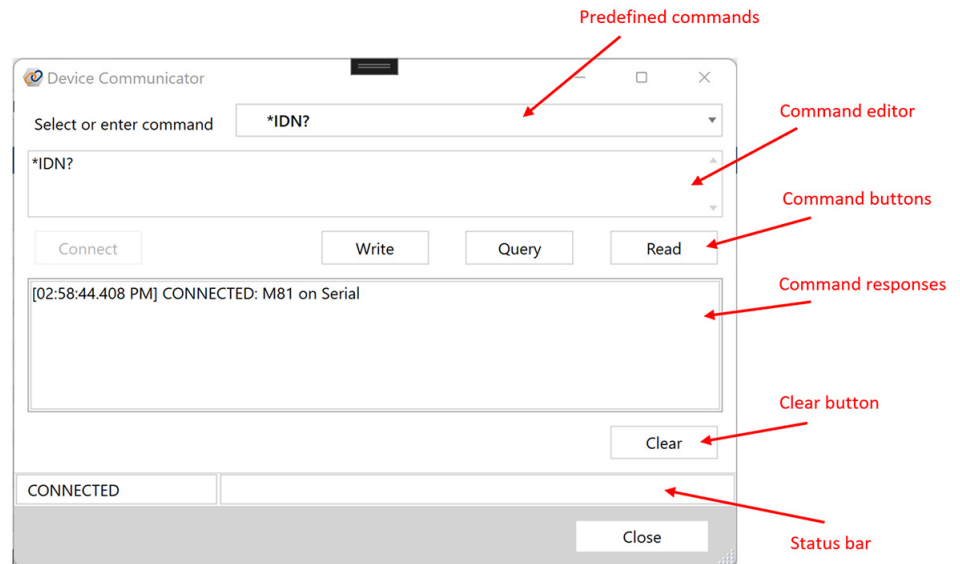


FIGURE 1-70 Device Communicator dialog box, showing controls

- **Predefined commands:** contains a list a commonly used predefined commands. To use one of these commands, select it from the list and it will be copied to the command editor control.
- **Command editor:** used to enter commands that should be sent to the connected device. The first entry in the predefined commands list is entered into the command editor by default. These commands can be entered from the keyboard or selected from the predefined commands, and can be modified prior to sending them to the device. If a command that is sent to the device is not found in the pre-

defined commands list, it is added to the top of the list so that it can be selected again without having to re-enter it.

- **Command buttons:** used to send a command to the connected device, or to read a response back from the device.
  - **Write button:** writes the contents of the command editor to the connected device. This command does not read back a response. If the command text is empty, this button is disabled.
  - **Query button:** performs a write followed by a read. Valid queries are commands that are expected to process the sent command, and then return a response from the connected device. A query usually ends with a question mark (?). If the command text is empty, this button is disabled.
  - **Read button:** reads back a response to a previous command from the connected device that is stored on the device. A timeout exception will result if the buffer is empty when this command is issued.
- **Command responses:** lists the sequence of commands that were sent to the connected device, as well as the responses that were returned. Entries are also prefixed with a time stamp, including millisecond resolution.
- **Clear button:** can be used to clear the contents of the responses box.
- **Status bar:** displays the status of the dialog box, including the connection status, and any relevant command status.

The following figure displays an example of a successful query (\*IDN?) of a connected device. Notice that the query was initiated by the Query button, and results in two entries in the response box: The first is a record of the query that was made, and the second is the response that was returned by the device.

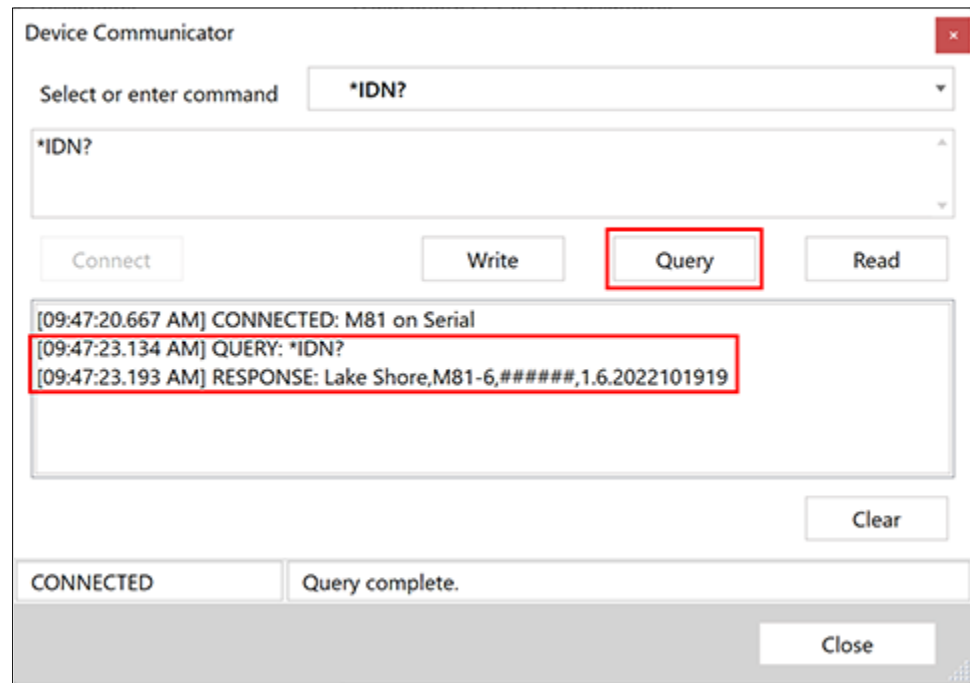


FIGURE 1-71 Device Communicator dialog box, showing a completed query

### 1.8.3.4 Process View

A Process View allows you to graphically display real-time indications for your instruments in common system configurations.

#### 1.8.3.4.1 Example process view

A Process View consists of a graphical representation of an experiment configuration overlaid with annotations for instrument indications. Each annotation consists of a description and value, and represents one instrument reading of interest.

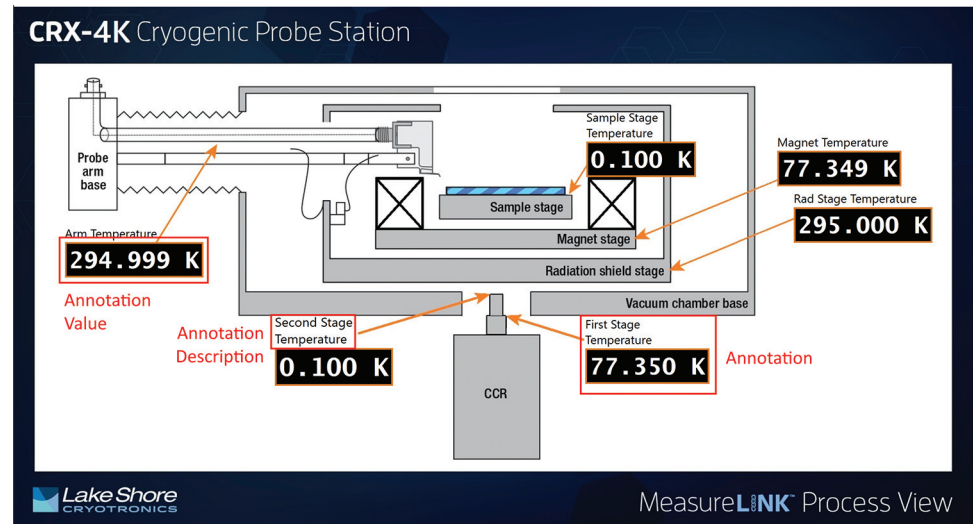


FIGURE 1-72 Process view

#### 1.8.3.4.2 Process View Driver

Process views can be added to a system configuration just like any other driver. To add a Process View, follow the same steps as for adding a driver (see section 1.8.3.2.1) and select the ProcessView driver from the Add Driver(s) pop-up window. You can add as many Process Views to your system configuration as needed.

The only setting required is selecting which process view file to display. Click **Browse** to find the file.



The driver and system configuration will be marked as invalid if no file is selected.

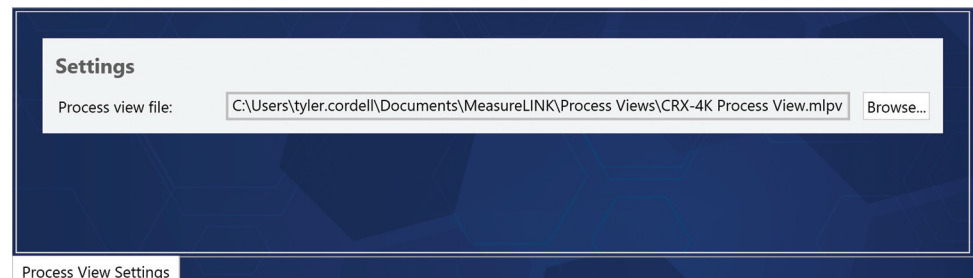


FIGURE 1-73 Process view settings

Once setup, the Process View will be displayed after the system configuration is applied.



The devices referenced in the Process View must also be included in the system configuration and be connected. Annotations for which values cannot be read will display an error.

## 1.8.4 User Preferences

The User Preferences pane allows the user to select preferred settings for using the application. Unless otherwise specified, these settings will be reflected throughout MeasureLINK™-MCS software, where applicable.

### 1.8.4.1 Significant Digits and Units

The user can select the number of significant digits for different values such as field values, temperatures, voltages, current values, and more. The number of significant digits for these values must be between 1 and 8.

Units can also be specified for field and temperature values. Supported field units include Gauss, Oersted, Tesla, and Ampere per meter. Supported temperature units include Kelvin and Celsius.



**If a connected instrument only supports one of these units, this preference will be ignored, and the instrument unit will be used for control and monitoring.**

Areas of MeasureLINK that do not implement user preferences include:

- Lake Shore CCR driver
- Keithley 2182 driver
- Lake Shore 336 driver
- Lake Shore 776 driver
- M81 Signal Analysis sequence step
- M81 Synchronous Measurement step

### 1.8.4.2 Usage Data

Lake Shore Cryotronics will automatically collect software usage data to help improve software quality. None of the information collected contains identifying information or measurement data, and any collected usage data will not be used for marketing or sales purposes.

The user can opt in and out of this data collection by clicking the checkbox.

### 1.8.4.3 Disabled Messages

Several MeasureLINK dialogs can be disabled so they are never displayed again. The user can reset this setting for all dialogs by clicking the button.

### 1.8.4.4 Default File Locations

These file locations are used as default directories throughout MeasureLINK when exporting files and data. Browse to select the preferred folders:

- *Documents folder*: the default folder where MeasureLINK related documents, such as sequence and script files, will be saved by default.
- *CSV folder*: the default location for exported and saved CSV files, such as those exported from charts.
- *Data folder*: used by some measurements as the default location for measurement data. This includes summary reports, data tables (as CSV files), and more. Measurements that currently use this setting include the M91 FastHall and MRS-



8400 Hall sequence steps. In scripts, users can use the “DataWriter” class to save data, which also defaults to the preferred Data folder.

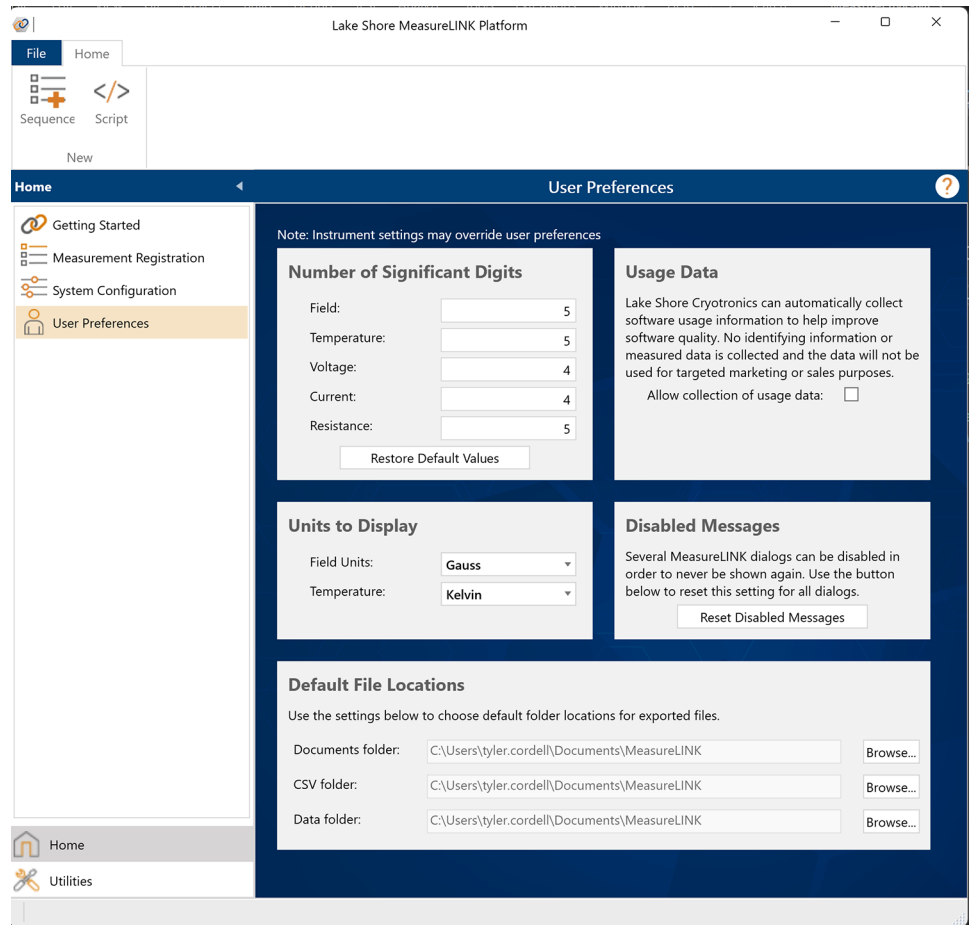


FIGURE 1-74 User Preferences

## 1.9 Utilities

Utilities allow the user to perform tasks on the currently selected instruments or on MeasureLINK™-MCS software overall. Some utilities are built into MeasureLINK, and others can be installed using application packs. The following sections describe the built-in utilities.

### 1.9.1 Application Packs

Different types of measurements require different instrument drivers, utilities, measurement scripts, and examples. MeasureLINK allows you to choose what types of measurements you wish to use and install only those components to MeasureLINK, using application packs. An application pack contains everything required for a certain type of measurement. You can add, update, or remove these application packs using the Application Pack utility.

The table below contains the current list of application packs. New and updated application packs may be downloaded from the Lake Shore website at <https://www.lakeshore.com/products/product-detail/measurelink/measurelink-mcs-application-packages>.

Application pack	Description
<b>Electrical</b>	Drivers, measurements, and examples for working with electrical instruments. Includes drivers for the Lake Shore M81 Synchronous Measurement System and the Lake Shore Model 155 source.
<b>Field platform</b>	Drivers, utilities, and examples for magnetic field control including the F71/F41 teslameters, permanent magnets, and the Lake Shore Model 643 and Model 648 power supplies.
<b>Janis cryostat</b>	Drivers and examples for working with a Janis Research superconducting cryostat, coupled with Lake Shore control hardware. Includes drivers for the Lake Shore Model 625 power supply and Model 336 temperature controller.
<b>M91 FastHall™</b>	Drivers, measurements, documentation, examples, and scripts for making Hall measurements with a MeasureReady™ M91 FastHall™ measurement controller.
<b>Temperature platform</b>	Drivers and examples for working with temperature control. Includes the Lake Shore Model 336 temperature controller driver.
<b>Quantum Design MultiVu™</b>	Drivers, utilities, and samples for taking measurements on the Quantum Design PPMS and DynaCool systems using the MultiVu™ software.

#### 1.9.1.1 Application Pack Utility Overview

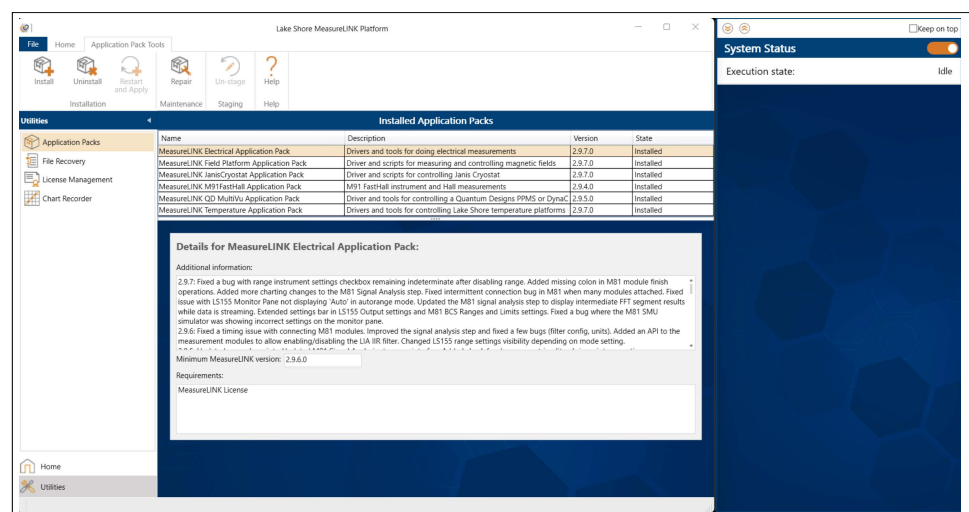


FIGURE 1-75 Application Packs

The Application Pack utility lists all currently installed application packs and their versions. Selecting an application pack will show detailed version information. Changes to the application packs can only take place when the MeasureLINK™-MCS software is starting up. Changes to application packs will take effect after restarting MeasureLINK by clicking the **Restart and Apply** button, which will launch the Application Pack installer. You may stage multiple application pack updates at one time. To cancel the staged operation, select the staged application pack and click the **Un-stage** button on the ribbon.

### 1.9.1.2 Installing Application Packs

To install a new application pack:

1. Press the **Install** button on the utility.
2. A dialog box appears. Choose the application pack you wish to install (application packs are files with an “.mlpack” extension), then click **Open**.
3. A dialog box appears, confirming the installation. Click **OK**.
4. The selected application pack is listed in bold type, showing that it is staged for installation when MeasureLINK is restarted using the **Restart and Apply** button.

Installed Application Packs - *Restart application to complete staged operations			
Name	Description	Version	State
MeasureLINK Field Platform Application Pack	Driver and scripts for measuring and controlling mag	0.5.0.0	Installed
MeasureLINK QD MultiVu Application Pack	Driver and tools for controlling a Quantum Designs F	0.1.0.0	Installed
<b>MeasureLINK M91FastHall Application Pack</b>	Driver and scripts for controlling M91 FastHall instrum	0.4.0.0	Installed on restart

FIGURE 1-76 Application pack staged for installation

### 1.9.1.3 Removing Application Packs

To remove an application pack:

1. Choose the application pack you wish to remove, then click the **Uninstall** button in the ribbon.
2. A dialog box appears, confirming the removal.
3. Press **OK**.
4. The selected application pack will be listed in strikethrough type, showing that it is staged for un-installation when MeasureLINK is restarted using the **Restart and Apply** button.

Installed Application Packs - *Restart application to complete staged operations			
Name	Description	Version	State
MeasureLINK Field Platform Application Pack	Driver and scripts for measuring and controlling magnetic fields	0.5.0.0	Installed
<del>MeasureLINK M91FastHall Application Pack</del>	<del>Driver and scripts for controlling M91 FastHall instrument</del>	<del>0.4.0.0</del>	<del>Uninstalled</del>
MeasureLINK QD MultiVu Application Pack	Driver and tools for controlling a Quantum Designs PPMS or DynaCool system	0.1.0.0	Installed

FIGURE 1-77 Application pack staged for un-installation

### 1.9.1.4 Repairing Application Packs

If an installed application pack is damaged or deleted, it can be repaired. To repair an application pack:

1. Choose the application pack you wish to repair, then click the **Repair** button on the ribbon.
2. A dialog box appears, confirming the repair.
3. Press **OK**.
4. The selected application pack will be listed in bold italic type, showing that it is staged for repair when MeasureLINK is restarted using the **Restart and Apply** button.

In a repair operation, the application pack contents are re-installed in MeasureLINK from a cached copy of the application pack.

Installed Application Packs - *Restart application to complete staged operations			
Name	Description	Version	State
MeasureLINK Field Platform Application Pack	Driver and scripts for measuring and controlling mag	0.5.0.0	Installed
MeasureLINK QD MultiVu Application Pack	Driver and tools for controlling a Quantum Designs F	0.1.0.0	Installed
<b><i>MeasureLINK M91FastHall Application Pack</i></b>	Driver and scripts for controlling M91 FastHall instrum	0.4.0.0	Repaired on restart

FIGURE 1-78 Application pack staged for repair

### 1.9.1.5 Staged Operations on Restart

When MeasureLINK™ is restarted with the **Restart and Apply** button, the Application Pack installer will perform the operations before relaunching the application. The status is displayed on the Application Pack installer screen.



FIGURE 1-79 MeasureLINK splash screen with application pack status

#### NOTE

**These operations may take a few minutes to complete.**

The following items can be installed by application packs (not all Applications Packs contain every type of content):

- *Instrument drivers and driver documentation:* installed drivers and documentation can be found on the Help page of the Backstage view (see section 1.3.6).
- *Measurements:* Installed measurements may be either sequence steps [insert link here] or registered measurement scripts (see section 1.7.2).
- *Utilities:* some drivers add additional utilities to the Utilities page of the MeasureLINK main window.
- *Examples:* example scripts demonstrating how to use a driver or feature can be found in the MeasureLINK folder on your hard drive: (C:\Users\Public\Documents\MeasureLINK\Examples). It is recommended that you make a copy of an example file before experimenting with it.

## 1.9.2 File Recovery

The File Recovery utility appears in the Utilities level. If the application (or computer) crashes, the working folders for all open files will remain in the working folder: \\Lake Shore MeasureLINK\OpenFiles. These files can be recovered using the File Recovery utility. File Recovery only appears if files are found that can be recovered.

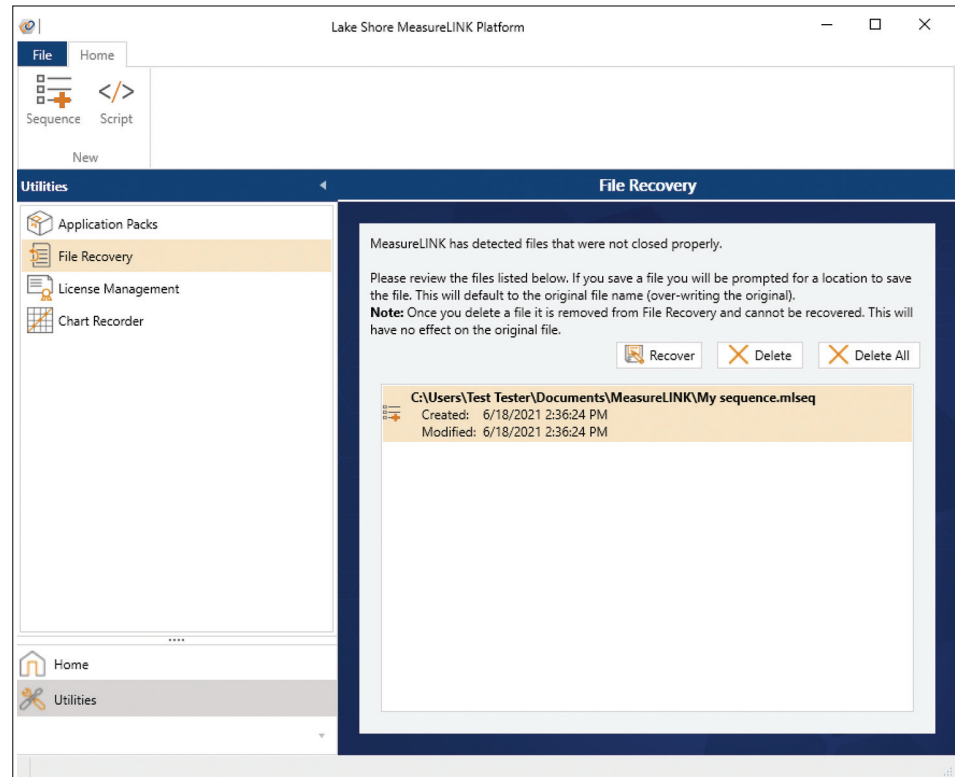


FIGURE 1-80 File recovery

When the software is restarted, the recovery utility lists files that were open when the application crashed. Select files from the list and choose either **Recover** or **Delete**:

- **Recover:** a Save As dialog box appears, allowing you to save the file to your desired location.
- **Delete:** a dialog box appears, confirming the deletion. Clicking **Yes** will the application will abandon the working folder.



**Delete will only delete the working file; not the original file.**

### 1.9.3 License Management

MeasureLINK™-MCS requires a user license in order to acquire data or connect to instruments. An activation code is required. If you don't have one, contact Lake Shore. An activation code can be used to run MeasureLINK on multiple computers, but the licenses will only be good for one year from the time of first activation. When the licenses expire, a new activation code will have to be obtained from Lake Shore. If the license expires you will be unable to run measurements. However, you will still be able to open and edit sequences and scripts.

It is also possible to evaluate MeasureLINK using an evaluation license. Evaluation licenses do not require an activation code but are only good for 90 days and can only be used once on a computer.

The current license status is displayed on the splash screen at application startup or can be viewed in the License Management utility within MeasureLINK. The License Management utility appears in the Utilities level.

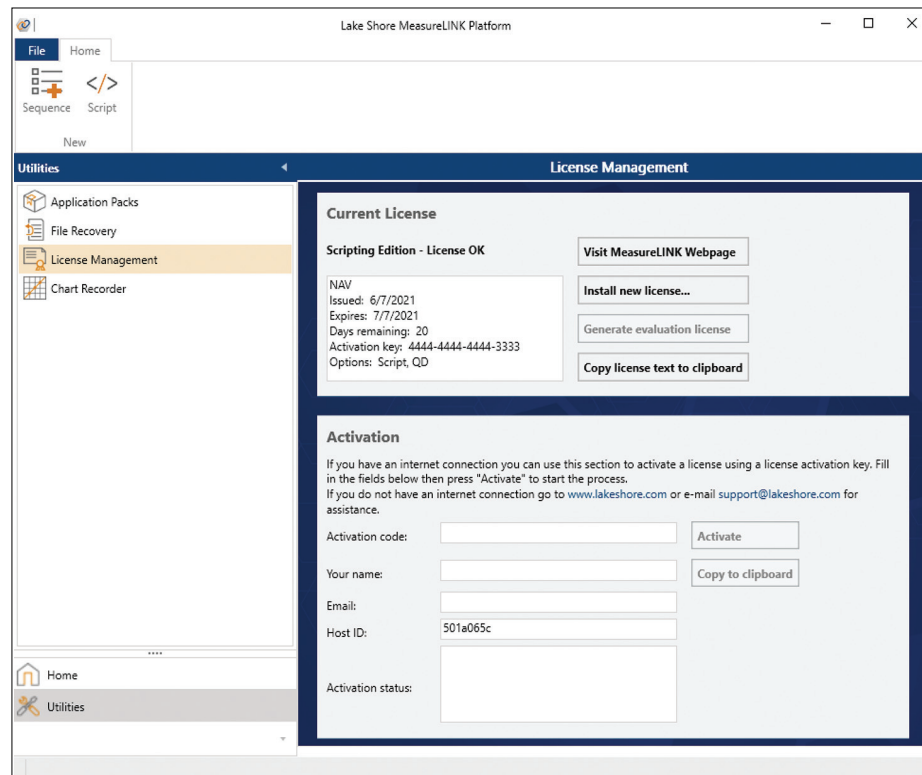


FIGURE 1-81 License Management



**If there is no valid license or the license is about to expire, the License utility will be shown automatically on the MeasureLINK main window.**

#### 1.9.3.1 Current License

The text box on the left shows the current license status and the known information about the license. Three buttons are provided:

- *Install new license*: used to install a license file downloaded or e-mailed directly from Lake Shore.

1. Click **Install new license...**
2. A dialog box appears.
3. Choose the license file and then click **Open**.
4. The license will be installed and applied to MeasureLINK.

- *Generate evaluation license*: used to create and install a 90-day evaluation license. This button is disabled if a license is already installed or if the evaluation license has expired. To generate a license:
  1. Click **Generate evaluation license**.
  2. The license will appear.
- *Copy license text to clipboard*: copies the contents of the license text box to the clipboard to use for record keeping or when contacting Lake Shore for support.

#### 1.9.3.2 Activation

The Activation group is used to obtain a license over the internet using an activation code. To obtain a license:

1. Provide an activation code, name, and valid e-mail address.
2. Click **Activate**.
3. MeasureLINK will send this data as well as the computer-specific Host ID to the activation server for validation. The new license will be installed and applied to MeasureLINK.
4. Status information is displayed in the Activation status text box during this process.
5. If no internet connection is available, use the **Copy to clipboard** button to copy the activation information to the clipboard. This can then be pasted into a file or e-mail to send to Lake Shore for manual activation.

## 1.9.4 Chart Recorder

The Chart Recorder utility allows you to display real-time data from your current instruments on one or more charts. All Chart Recorder charts plot the data value versus time. Device drivers must specifically support the Chart Recorder. Currently, not all drivers support the Chart Recorder, but more will be supported in future MeasureLINK versions.

### 1.9.4.1 Chart Recorder Overview

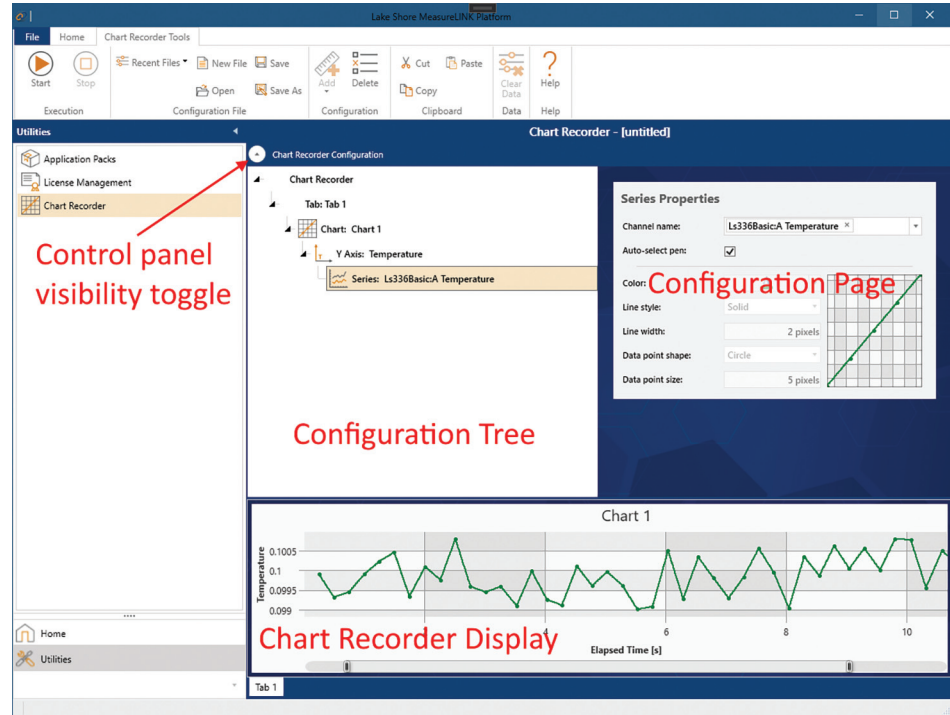


FIGURE 1-82 Chart Recorder

The Chart Recorder has a configuration panel and a chart display panel. The configuration panel is used to add elements to the Chart Recorder. This panel can be shown or hidden at any time, using the round button at the upper left of the Chart Recorder. Collapsing the configuration panel leaves more room for displaying the Chart Recorder chart(s).



### 1.9.4.2 Chart Recorder Configuration Panel

The configuration panel is split vertically into two areas. The left side of the screen is the Configuration tree, and the right is the property page for the currently selected item in the tree. The Chart Recorder uses a hierarchical tree model where each item in the tree “owns” the items below it. For example, the chart can contain one or more tabs, each of which contains one or more charts. Each chart contains one or more axes, and so on. The chart elements are:

Element	Description
<b>Chart Recorder</b>	The “root” of the tree. This element cannot be deleted or added.
<b>Tab</b>	The Chart Recorder can display charts on multiple tabs. Tab properties control the name and layout of the charts on the tab.
<b>Chart</b>	Charts can contain multiple Y axes, each of which can contain multiple series. Only one X axis is supported, and it is always Elapsed Time. Chart properties control data logging, the title, and the legend for the chart.
<b>Axis</b>	An axis hosts one or more series. Axis properties control the Title, Data type (log/linear), and position of the axis.
<b>Series</b>	Each series displays a data channel from an instrument. Series properties include the selected Channel, series name and appearance of the series on the chart.

Some of the element properties can only be changed when the Chart Recorder is not collecting data, others cannot be changed once other elements are added.

Chart Recorder elements are added and deleted using the ribbon controls. For example, to add a new chart to a tab:

1. Select the tab.
2. Click the **Add** button on the ribbon.
3. Click **Add chart**.

Clipboard operations are also supported on elements. Each Chart Recorder element is described in more detail below.

#### 1.9.4.2.1 Chart Recorder root element

The Chart Recorder element has no real properties to set. Instead it displays a very short getting started guide to the Chart Recorder.

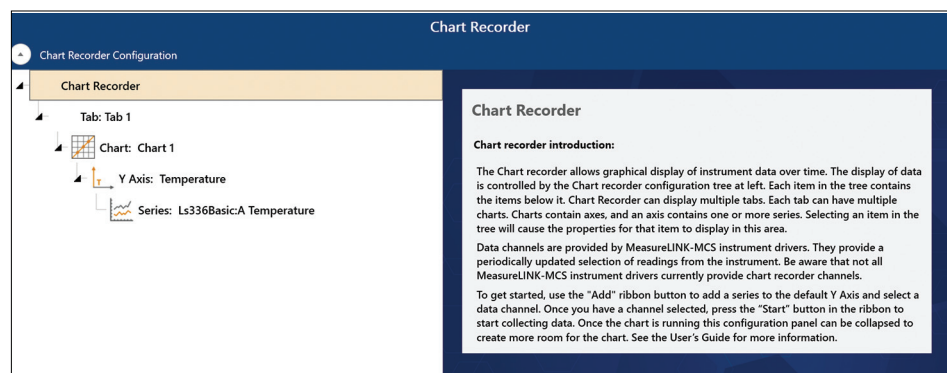


FIGURE 1-83 Chart Recorder root element

### 1.9.4.2.2 Tab element

The tab properties allow you to set the title that will display on the tab as well as how charts will be arranged on that tab. Three options are supported:

Element	Description						
<b>Horizontal row</b>	All charts are in a horizontal row on the tab.						
<b>Vertical column</b>	Charts are stacked in a column, from top to bottom.						
<b>Grid</b>	Charts are arranged in rows and columns. Each row is two charts wide and rows are added as needed to hold all the charts. So if you have five charts they will be arranged like this: <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>1</td> <td>2</td> </tr> <tr> <td>3</td> <td>4</td> </tr> <tr> <td>5</td> <td></td> </tr> </table>	1	2	3	4	5	
1	2						
3	4						
5							

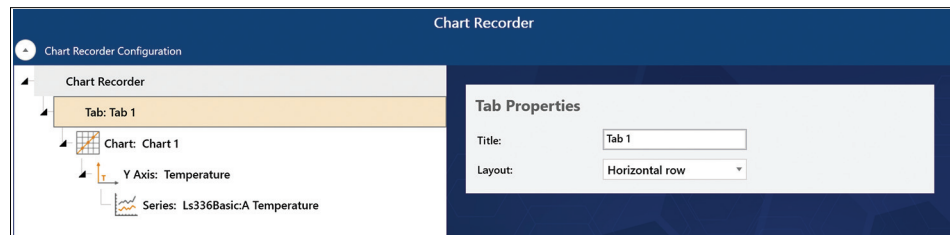


FIGURE 1-84 Tab element

### 1.9.4.2.3 Chart element

The chart properties allow you to control how the chart appears and where data is logged.

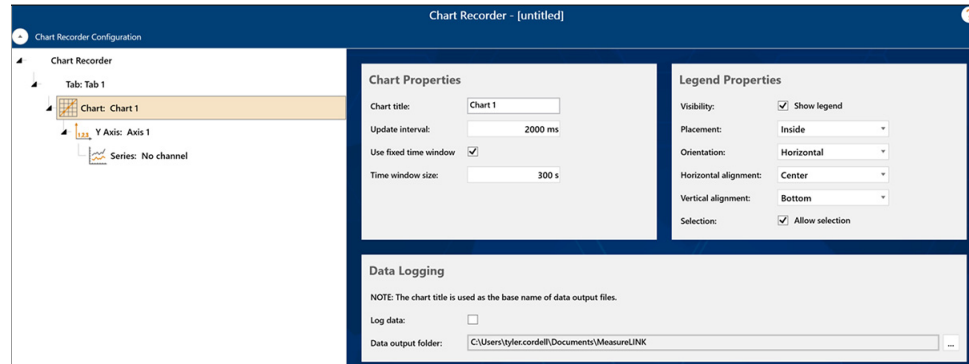


FIGURE 1-85 Chart element

These controls are arranged into three groups:

- **Chart Properties:**
  - **Chart title:** the title that appears on the top of the chart. It can be blank or any text you like.
  - **Update interval:** the interval at which chart data is updated and collected.
  - **Time window:** as the Chart Recorder runs, more data is added to the chart and the X axis will get longer. A time window makes the data visible and manageable. With a time window, all data will be on the chart but only data in the time window is visible. The chart will scroll to keep the latest data visible. To see all data, double-click the chart.
- **Legend Properties:** Optionally, the chart can show a legend describing all the series on the chart. This legend will only appear if there is more than one series on the chart. Use the fields in this group to control the placement and orientation of

the legend. If the Allow selection checkbox is selected, each series in the legend will contain a checkbox that will hide or show that series.

- **Data Logging:** The Chart Recorder can log collected data to .csv files. To log data, check the checkbox and select the folder where you want to store the data. Each chart produces a separate set of .csv files. Files are named based on the chart title. A new file is created every time the Chart Recorder is started, stopped, or the series in the chart change.

#### 1.9.4.2.4 Y-axis Properties

The series properties control the appearance and data type of a vertical axis on the chart.

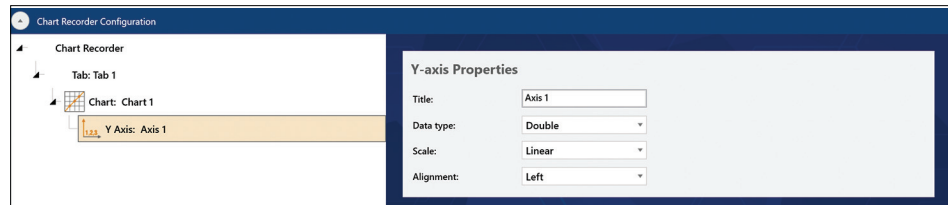


FIGURE 1-86 Y-axis Properties

The properties are:

- **Title:** appears next to the axis on the chart. It can be blank or custom text.
- **Data type:** data type of the axis. Because this field also controls the data type of any series on this axis, the value cannot be changed once a series is added.

Supported data types are:

Element	Description
Double	Floating-point decimal numbers
Field	Field values
Integer	Whole integer numbers
Temperature	Temperature values

- **Scale:** select linear or logarithmic scale. Negative numbers do not work for logarithmic data; this value cannot be changed once data is on the chart.
- **Alignment:** choose left or right side placement for the axis.

#### 1.9.4.2.5 Series Properties

The series properties control the data in a series and the appearance of series on the chart.

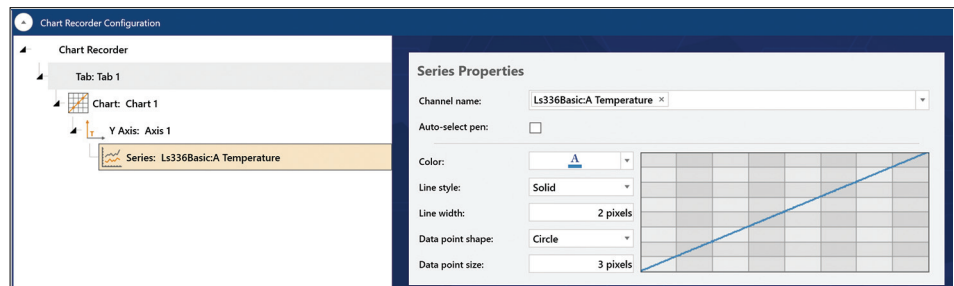


FIGURE 1-87 Series Properties

### Selecting a channel

Each series in a Chart Recorder chart collects data from a data channel. The available data channels depend on the instruments in the active system configuration. Not all instruments currently expose data channels. The series can only show channels that have the same data type as the axis where the series is added. For example, a temperature axis can only show temperature channels.

The Channel name field filters and selects a channel. Click the drop-down icon on the right to show a list of all data channels of the correct data type.

Name	Data Type	Description
Ls336Basic:A Temperature	Temperature	Measured temperature
Ls336Basic:A Set point temperature	Temperature	Temperature setpoint
Ls336Basic:B Temperature	Temperature	Measured temperature
Ls336Basic:B Set point temperature	Temperature	Temperature setpoint
Ls336Basic:C Temperature	Temperature	Measured temperature
Ls336Basic:C Set point temperature	Temperature	Temperature setpoint
Ls336Basic:D Temperature	Temperature	Measured temperature
Ls336Basic:D Set point temperature	Temperature	Temperature setpoint

FIGURE 1-88 Channel name

Within the drop-down list, sort the channels by clicking the column headers. Or filter the channels using the filter icon in the column header.

The screenshot shows a dropdown menu for the 'Channel name' field. The selected channel is 'Ls336Basic:A Temperature'. The dropdown list contains the following items:

- Select All
- Ls336Basic:A Set point temperature
- Ls336Basic:A Temperature
- Ls336Basic:B Set point temperature
- Ls336Basic:B Temperature
- Ls336Basic:C Set point temperature
- Ls336Basic:C Temperature
- Ls336Basic:D Set point temperature
- Ls336Basic:D Temperature

Below the list is a filter section with the following options:

- Show rows with value that
- Is equal to
- And
- Is equal to

Buttons for 'Filter' and 'Clear Filter' are located at the bottom of the filter section.

FIGURE 1-89 Filtering channels

Do a quick search by typing a search string in the Channel name field, which will automatically display the drop-down and filter the list to items that contain the text entered.

Channel name: set		
Name	Data Type	Description
Ls336Basic:A Set point temperature	Temperature	Temperature setpoint
Ls336Basic:B Set point temperature	Temperature	Temperature setpoint
Ls336Basic:C Set point temperature	Temperature	Temperature setpoint
Ls336Basic:D Set point temperature	Temperature	Temperature setpoint

FIGURE 1-90 Searching channels

### Controlling series appearance

Each new series in a chart is assigned a unique pen, which is auto-selected by default. An example of how the series will display is shown in the properties window. To customize the appearance of the pen, uncheck the Auto-select pen checkbox. Custom fields include color, marker shape, and sizes/widths. The assigned pen is retained and can be replaced by checking Auto-select pen again.

### 1.9.4.3 Saving and Restoring Chart Recorder Configurations

A Chart Recorder configuration can be saved and restored using the tools in the Configuration File group on the Chart Recorder Tools ribbon. Chart Recorder configuration files are saved to the user files folder by default using the file extension “.mlrc.”

The currently selected configuration file name is displayed in the Chart Recorder window. If a file name has not been assigned, the name is displayed as “untitled.”

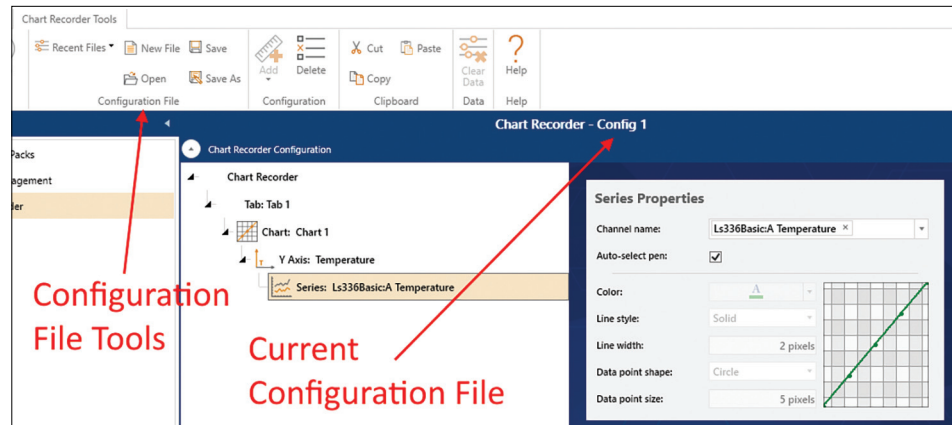


FIGURE 1-91 Chart Recorder root element

The tools in this group perform the following functions:

#### New File

Sets the Chart Recorder configuration back to the default, targeting a new, untitled configuration file. If the current configuration has been modified, the user will be prompted to save it prior resetting to the default.

#### Open

Displays the Open dialog box, allowing the user to open an existing Chart Recorder configuration file. If the current configuration has been modified, the user will be prompted to save it prior opening the selected file.

#### Save

Saves any modifications to the current file. If a file name has not been chosen, the user will be prompted to create one.

#### Save As

Displays the Save As dialog box, allowing the user to save the current configuration to a new file.

#### Recent Files

Displays a drop down list of recently-used configuration files. The list of files is sorted with the most-recently used file at the top. If the current configuration has been modified, the user will be prompted to save it prior to opening the selected file.

# Chapter 2: Remote Operation

## 2.1 General

This chapter explains remote operation for the Lake Shore MeasureLINK™-MCS software. Remote operation allows third-party software to integrate with and control the MeasureLINK™-MCS software.

## 2.2 Selecting and Configuring an Interface

The sections below describe the process to select and configure your remote interface.

### 2.2.1 TCP-IP Interface

When MeasureLINK is running, a TCP communication server is exposed using the IP address assigned by network using the port that is defined in the PlatformConfiguratin.json configuration file. The default port is: "RemotePortNumber": 7777. In order to communicate with this server properly, clients should enable the termination character feature.

### 2.2.2 Enabling Remote Operation in MeasureLINK™

In order to enable remote operation, MeasureLINK should be started using the "/R" command line argument. This can be accomplished using the following steps:

1. Right-click on the MeasureLINK shortcut icon and select **Properties**.
2. On the **Shortcut** tab of the Properties dialog box, place the cursor at the end of the string of characters in the Target field, outside of the quotes.
3. Type a single space and then add the "/R" switch (without the quotes).
4. Click **Apply** to save the changes.
5. Click **OK** to exit.

## 2.3 SCPI Language

All commands and queries used to communicate with MeasureLINK are formatted using the SCPI (Standard Commands for Programmable Instruments) standard. The complete SCPI specification can be found on the IVI website: <http://www.ivifoundation.org/docs/scpi-99.pdf>.

## 2.4 Common Commands

Common commands are commands that are common to all devices. MeasureLINK supports the following:

Mnemonic	Name	Description	Ref
*IDN?	Identification query	Returns the manufacturer, model number, serial number, and version of the unit.	

TABLE 2-1 Common commands

## 2.5 Measurement Commands

The measurement commands and queries are arranged into two subsystems, the Fetch subsystem and the Source subsystem.

### 2.5.1 Fetch Subsystem

Mnemonic	Description	Ref
FETCh:FIELD? [deviceName]	Queries the current field reading (Tesla)	a
FETCh:TEMPerature? [deviceName]	Queries the current temperature (Kelvin)	b

TABLE 2-2 *Fetch subsystem*

- a. FETCh:FIELD? [deviceName]  
[deviceName] = optional parameter indicating which device should be queried. If this parameter is omitted, the first valid device will be queried.

This query returns the current field reading in Tesla units. In order for this query to be successful, the following conditions must be met:

1. The system configuration must contain at least one device that is of type FieldController
2. At least one FieldController device must be connected as indicated by the Connected switch on the Monitor window.

- b. FETCh:TEMPerature? [deviceName]  
[deviceName] = optional parameter indicating which device should be queried. If this parameter is omitted, the first valid device will be queried.

This query returns the current temperature reading in Kelvin units. In order for this query to be successful, the following conditions must be met:

1. The system configuration must contain at least one device that is of type TemperatureController.
2. At least one TemperatureController device must be connected as indicated by the Connected switch on the Monitor window.



**If the conditions are not met, a “HardwareMissing” error will be added to the SCPI error queue.**

### 2.5.2 Source Subsystem

Mnemonic	Description	Ref
SOURce:FIELD:STATe? [deviceName]	Queries the current field control state	a
SOURce:FIELD:SETPoint <NRf>,[deviceName]	Sets the field control setpoint (Tesla)	b
SOURce:FIELD:SETPoint? [deviceName]	Queries the field control setpoint (Tesla)	c
SOURce:TEMPerature:STATe? [deviceName]	Queries the current temperature control state	d
SOURce:TEMPerature:SETPoint <NRf>,[deviceName]	Sets the temperature control setpoint (Kelvin)	e
SOURce:TEMPerature:SETPoint? [deviceName]	Queries the temperature control setpoint (Kelvin)	f

TABLE 2-3 *Source subsystem*

- a. SOURce:FIELD:STATe? [deviceName]  
[deviceName] = optional parameter indicating which device should be queried. If this parameter is omitted, the first valid device will be queried.

This query returns the current field control state. Possible responses include Off, Ramping, Settling, OK and Fault. In order for this query to be successful, the following conditions must be met:

1. The system configuration must contain at least one device that is of type FieldController.



2. At least one FieldController device must be connected as indicated by the Connected switch on the Monitor window.

- b. `SOURce:FIELD:SETPoint <Nrf>, [deviceName]`  
 <Nrf> = desired field control setpoint in Tesla  
 [deviceName] = optional parameter indicating which device should be set. If this parameter is omitted, the first valid device will be set.

This command sets the field control setpoint. In order for this command to be successful, the following conditions must be met:

1. The system configuration must contain at least one device that is of type FieldController.
2. At least one FieldController device must be connected as indicated by the Connected switch on the Monitor window.

- c. `SOURce:FIELD:SETPoint? [deviceName]`  
 [deviceName] = optional parameter indicating which device should be queried. If this parameter is omitted, the first valid device will be queried.

This query returns the current field control setpoint in Tesla. In order for this command to be successful, the following conditions must be met:

1. The system configuration must contain at least one device that is of type FieldController.
2. At least one FieldController device must be connected as indicated by the Connected switch on the Monitor window.

- d. `SOURce:TEMPerature:STATE? [deviceName]`  
 [deviceName] = optional parameter indicating which device should be queried. If this parameter is omitted, the first valid device will be queried.

This query returns the current temperature control state. Possible responses include Unknown, Off, Ramping, Paused, Waiting, Settling, and Settled. In order for this query to be successful, the following conditions must be met:

1. The system configuration must contain at least one device that is of type TemperatureController.
2. At least one TemperatureController device must be connected as indicated by the Connected switch on the Monitor window.

- e. `SOURce:TEMPerature:SETPoint <Nrf>, [deviceName]`  
 <Nrf> = desired temperature control setpoint in Kelvin.  
 [deviceName] = optional parameter indicating which device should be set. If this parameter is omitted, the first valid device will be set.

This command sets the temperature control setpoint. In order for this command to be successful, the following conditions must be met:

1. The system configuration must contain at least one device that is of type TemperatureController.
2. At least one TemperatureController device must be connected as indicated by the Connected switch on the Monitor window.

- f. `SOURce:TEMPerature:SETPoint? [deviceName]`  
 [deviceName] = optional parameter indicating which device should be queried. If this parameter is omitted, the first valid device will be queried.

This query returns the current temperature control setpoint in Kelvin. In order for this command to be successful, the following conditions must be met:

1. The system configuration must contain at least one device that is of type TemperatureController.
2. At least one TemperatureController device must be connected as indicated by the Connected switch on the Monitor window.



**If the conditions are not met, a “HardwareMissing” error will be added to the SCPI error queue.**