NI-DCPower User Manual

2025-03-26

Contents

NI-DCPower User Manual

The NI-DCPower User Manual provides detailed descriptions of the product functionality and the step by step processes for use.

Looking for Something Else?

For information not found in the User Manual for your product, such as specifications and API reference, browse ***Related Information***.

Related information:

- Download NI-DCPower
- License Setup and Activation
- NI-DCPower Release Notes
- NI-DCPower LabVIEW Reference
- NI-DCPower C Function Reference
- NI-DCPower C# .NET Reference
- NI-DCPower Python Reference

NI-DCPower Overview

The NI-DCPower instrument driver supports NI power supplies, source measure units (SMUs), and electronic loads (E-loads). This User Manual provides detailed descriptions of the driver functionality.

Note This User Manual refers to power supplies, SMUs, and E-loads inclusively as instruments. If information pertains only to a specific instrument, the User Manual refers to the instrument type.

NI-DCPower New Features and Changes

Learn about updates, including new features and behavior changes, introduced in each version of NI-DCPower.

Discover what is new in the latest releases of NI-DCPower.

> **Note** If you cannot find new features and changes for your version, it might not include user-facing updates. However, your version might include non-visible changes such as bug fixes and compatibility updates. For information about non-visible changes, refer to your product ***Release Notes***.

Related information:

- NI-DCPower Release Notes
- Software and Driver Downloads

NI-DCPower 2025 Q1 New Features and Changes

- Support for INHIBIT pin added to the following modules:
  - PXIe-4051
  - PXIe-4150
  - PXIe-4151

NI-DCPower 2024 Q3 New Features and Changes

- Support for PXIe-4162/4163 Merged Channels
- Support for PXIe-4150

Related information:

- PXIe-4162 Merged Channels
- PXIe-4163 Merged Channels

NI-DCPower 2024 Q1 New Features and Changes

- Added Accessory Detection support for ACC-4162-3 and ACC-4163-3

NI-DCPower 2023 Q4 New Features and Changes

- Support for OpenSUSE 15.5
- Dropped support for OpenSUSE 15.3
- Support for PXIe-4151
- Support for PXIe-4051
  - Added programmable current level slew rate support for PXIe-4051
  - Enabled setting Conduction Voltage threshold for PXIe-4051
  - New mode supported by the Instrument Mode property: `E_LOAD`
- Enhanced Output Cutoff feature for PXIe-4135/4136/4137/4138/4139
- Accessory Detection support for ACC-4162-1 and ACC-4163-1

NI-DCPower 2023 Q3 New Features and Changes

- Support for PXIe-4162 Asymmetrical Compliance Limit
- Support for PXIe-4162 Measurement Autorange

Updates and Changes for NI-DCPower Extended Support Versions

Browse updates and changes made in NI-DCPower versions on extended support.

> **Note** If you cannot find changes for your version, it might be a more recent version, documented as a new feature. Or, your version might not have included user-facing updates. You can find more information about non-visible changes, such as bug fixes, compatibility updates, and stability adjustments or maintenance adjustments, in the product ***Release Notes***, available on ni.com.

NI-DCPower 2023 Q2 New Features and Changes

- LCR Source Aperture Time attribute for PXIe-4190
- Customizable DC Bias Transient Response for PXIe-4190
- Detection of unbalanced conditions on PXIe-4190

NI-DCPower 2023 Q1 New Features and Changes

- Support for PXIe-4163 Remote Sense Over Voltage Protection (OVP)
- Support for PXIe-4162 Remote Sense Over Voltage Protection (OVP)
- Support for PXIe-4190 DC Bias Automatic Level Control Off
- Support for PXIe-4190 "As Configured" as an LCR Open/Short/Load Compensation Data Source
- Support for PXIe-4190 SMU Measurement Autorange

NI-DCPower Examples

NI installs example code with your software or driver that demonstrates the functionality of NI-DCPower. Use these examples to learn about the product or accelerate your own application development.

Most NI products install examples that you can access directly or from within NI software. The example experience can differ slightly across products and versions.

Installed Example Locations

Option		Installed Example Locations
LabVIEW		`<LabVIEW>\examples\instr\nidcpower`, where `<LabVIEW>` is the LabVIEW directory for the specific LabVIEW version you installed on your system.
LabWindows/CVI		`Users\Public\Documents\National Instruments\CVI\samples\niDCPower`
.NET	4.0	`Users\Public\Documents\National Instruments\NI-DCPower\Examples\DotNET 4.0`
	4.5	`Users\Public\Documents\National Instruments\NI-DCPower\Examples\DotNET 4.5`

Common NI-DCPower Examples

NI-DCPower Example	Description
NI-DCPower Source DC Voltage	Demonstrates how to force an output voltage.
NI-DCPower Source DC Current	Demonstrates how to force an output current.
NI-DCPower Hardware-Timed Voltage Sweep	Demonstrates how to sweep the voltage on a single channel and display the results in a graph.
NI-DCPower Measure Record	Demonstrates how to take multiple measurements in succession.
NI-DCPower Measure Step Response	Demonstrates how to measure the output while it is changing.

Browsing and Searching for Examples in NI Example Finder

Use NI Example Finder to browse and to search for examples.

You can use NI Example Finder to find examples for the following products.

- LabVIEW
- LabWindows/CVI
- NI drivers accessible from LabVIEW
- NI drivers accessible from LabWindows/CVI

1. Launch LabVIEW or LabWindows/CVI.
2. Open NI Example Finder.

Option	Description
LabVIEW	Select **Help** » **Find Examples.** from the menu bar.
LabWindows/CVI	Click **Find Examples...** from the Examples section of the Welcome Page.

   NI Example Finder launches.
3. **Optional:** Configure NI Example Finder for LabWindows/CVI.
   a. Click **Setup**. Configure Example Finder opens.
   b. In Configure Example Finder, click **Software**, then select LabWindows/CVI, and click **OK**.

   NI Example Finder updates with all the examples for LabWindows/CVI.
4. Search the example VIs for your product.

Option	Description
Click the Browse tab.	Choose **Browse** when you want to drill down through folders to find examples organized by task category.

Option	Description
	Tip Examples installed with NI drivers or third-party drivers are often found within the **Hardware Input and Output** folder. Examples installed with toolkits or modules are often found within the **Toolkits and Modules** folder.
Click the Search tab.	Choose **Search** when you want to find examples by searching for topics, products, or modules relevant to your application.

5. To open an example, double-click the folder or the example.

Tip You can modify an example VI to fit your application. You can also copy and paste from one or more examples into a VI that you create.

Using Measurement & Automation Explorer for NI-DCPower

Use Measurement & Automation Explorer (MAX) to complete the following common tasks for NI-DCPower.

Running the Test Panels

Use test panels to interactively test the functionality of your device. To run the test panels, right-click the device name in the MAX configuration tree, and select **Test Panels**.

Removing Your Device

To remove the device from your configuration, right-click the device name in the configuration tree and select **Delete**. This option is only valid if the physical device is no longer present in the system.

Viewing or Changing Device Properties

To view or change device properties, right-click the device name in the configuration tree and select **Properties**.

Using InstrumentStudio with Your NI-DCPower Instrument

You can monitor, control, and record measurements from supported NI-DCPower instruments using InstrumentStudio. Use InstrumentStudio to perform interactive measurements on several different device types, including instruments, in a single program.

Accessing InstrumentStudio

InstrumentStudio is installed with NI-DCPower. You can access InstrumentStudio in one of the following ways:

- From the Start menu. InstrumentStudio launches with a soft front panel that is populated with devices that are detected on your system.
- From Measurement & Automation Explorer (MAX), select a device and then click **Test Panels...**. InstrumentStudio launches with a soft front panel for the device you select.

Related information:

- InstrumentStudio Overview

Programming with NI-DCPower

NI-DCPower, an Interchangeable Virtual Instrument (IVI)–compliant instrument driver, is included with your NI instrument and communicates with all NI programmable instruments. NI-DCPower features a set of operations and properties that exercise the functionality of the instrument.

Related information:

- NI-DCPower LabVIEW Reference
- NI-DCPower C Function Reference
- NI-DCPower C# .NET Reference
- NI-DCPower Python Reference

Programming Flow

Complete the following steps to program your instrument.

Related concepts:

- Programming States
- Single Point Source Mode
- Sequence Source Mode
- Advanced Sequencing
- Pulsing

Opening a Session

To open a session, call NI-DCPower Initialize With Independent Channels.

For any application you write, you must open a session to establish communication with the specified device(s) or channel(s) by initializing.

Initializing returns an instrument handle with the session configured to a known state. Initialization can take a significant amount of time compared to other NI-DCPower functions, so you should not include it in a loop when repeatedly acquiring data.

Ideally, your program should call NI-DCPower Initialize With Independent Channels one time. If the `reset` parameter is set to **TRUE**, device channels are reset to the default state, which may include resetting relays.

Initiating Generation and Acquisition

To apply the configuration and start generation, use NI-DCPower Initiate With Channels.

Measuring, Querying, or Fetching

The source mode that you select for configuring channels determines how NI-DCPower acquires measurements.

To acquire measurements in single point source mode complete the following steps.

1. Measure with NI-DCPower Measure Multiple.
2. To query the output state, call NI-DCPower Query In Compliance.

NI-DCPower automatically acquires measurements when you configure the following functions:

- NI-DCPower Create Advanced Sequence With Channels
- NI-DCPower Set Sequence
- NI-DCPower Configure Output Function set to **Pulse Voltage** or **Pulse Current**

NI-DCPower automatically acquires the measurements by coercing the NI-DCPower Measure When property to `Automatically After Source Complete`. NI-DCPower returns the measurement values in an array.

Tip If you want the measure unit to operate independently of the source unit in this context, set the NI-DCPower Measure When property to a value other than `Automatically After Source Complete`.

Closing the Session

To close a session, use NI-DCPower Close.

Closing a session is essential for freeing resources, including deallocating memory, destroying threads, and freeing operating system resources. Ensure that you close every session that you initialize, even if an error occurs during the program. When debugging your application, it is common to abort execution before you close.

Note Aborting the execution is not likely to cause problems, but NI does not recommend doing so.

The channels continue to operate in their last configured state when you close a session. If you close a session while the output channels are enabled and actively sourcing or sinking power, the channels continue to source or sink power until you disable or reset them.

Configuring Channels

Channels have two basic source modes: ***Single point*** and ***sequence***. The sequence source mode encompasses both simple sequences and advanced sequences. In each of these modes, you can output constant or pulsed voltage or current.

Note

- Attribute values do not persist between sessions. If you close a session and open a new session, all attributes assume their default values. However, the default values are not committed to hardware until the channels enter the Committed or Running state.
- Some NI-DCPower instruments have resources that are shared by different sessions to the same instrument. You cannot configure these resources independently by channel.

Configuring Channels in Single Point Source Mode

Complete the following steps to configure channels in single point source mode.

1. Set the `Source Mode` to **Single Point** using NI-DCPower Configure Source Mode With Channels.

2. Configure the source unit.
3. (Optional) Configure the measure unit.

Configuring Channels in Sequence Source Mode

Complete the following steps to configure channels in sequence source mode.

1. Set the `Source Mode` to **Sequence** using NI-DCPower Configure Source Mode With Channels.
2. Configure the source unit.
3. Create a simple sequence with NI-DCPower Set Sequence, or create an advanced sequence with NI-DCPower Create Advanced Sequence With Channels.
4. (Optional) Configure the measure unit.

Using Properties and Attributes

NI-DCPower contains high-level VIs and functions that set most of the properties and attributes in the NI-DCPower API.

Some properties and attributes are not accessible through the high-level VIs and functions. You must set the values for these properties and attributes using the appropriate property or attribute.

Accessing Properties in LabVIEW

In LabVIEW, properties are accessed through the NI-DCPower property node. To access properties in LabVIEW, complete the following steps:

1. Open a VI.
2. In the block diagram view, navigate to the **NI-DCPower** palette.
3. Add the property node icon to the block diagram.
4. Left-click the property node, and select the property that you want to use.
5. To add additional properties, resize the property node. To resize the property node, drag the resizing handle at the top or bottom of the node and release the mouse button.

When to Use an Active Channel

The Active Channel property defines the channels that channel-based properties apply to. The Active Channel is listed first in the property node.

You must pass an Active Channel in any of the following cases:

- For multichannel sessions, if you want to configure the channels differently from one another.
- For multichannel sessions, when reading a property where multiple channels may have different values for that property.
- For multichannel sessions where the channels span multiple physical instruments, if you are reading any instrument-based properties; in this case, pass the instrument name to Active Channel rather than a channel name.

You do not need to pass an Active Channel, or can pass an empty string, in any of the following cases:

- If the properties you are using are neither channel-based nor instrument-based
- If your session includes only one channel
- If you want to configure all channels in a multichannel session identically
- If you want to read a property from all channels when all channels have the same property value or from an instrument-based property

Accessing Attributes

In C and Visual Basic 6.0, access the attributes with the Get Attribute and Set Attribute functions. The Get and Set Attribute functions exist for each supported data type in NI-DCPower.

Setting Properties and Attributes Before Reading Them

Properties and attributes are modified when you set them or when you call a configuration VI or function that sets them, respectively. It is important to set the properties or attributes or call any configuration VIs or functions before reading back any property or attribute values for the following reasons:

- Values read are coerced depending on the current configuration of the session. If you read a property or attribute value and then set other properties or attributes, the value read may no longer be valid.
- The driver verifies that the configuration of the session is valid at the time the property or attribute is read. It is possible to get an error when reading a property or attribute if the configuration is not valid at that point, even when a setting later could make it valid.
- Reading properties or attributes causes the driver to verify the current configuration. If you change some of the settings later, those settings need to be validated again.

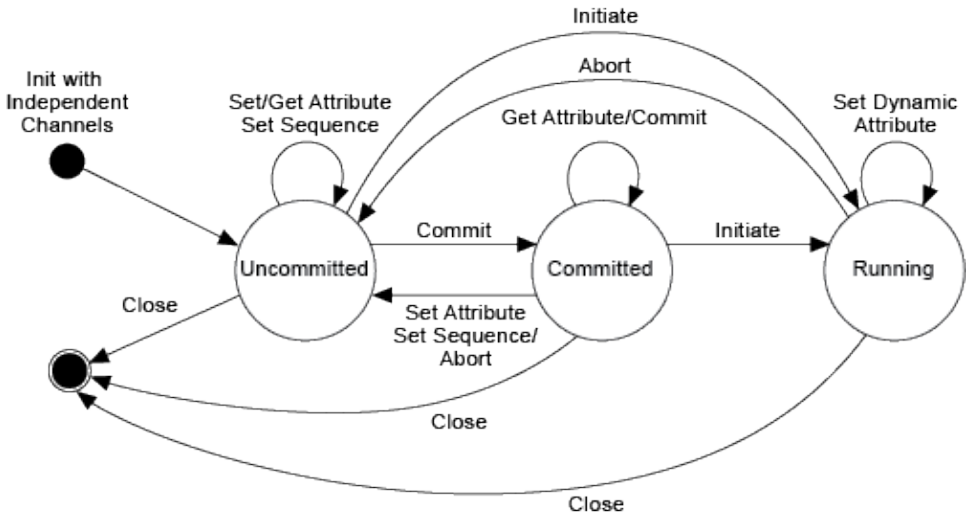
Programming States

An NI-DCPower channel has three main states:

- Uncommitted
- Committed
- Running

Each channel in a session can move through these states independent of one another.

Figure 1. NI-DCPower Programming States



Note Sessions created using the Initialize With Independent Channels function have different capabilities from those created using prior NI-DCPower initialize functions.

Uncommitted

Channels enter the Uncommitted state when you call NI-DCPower Initialize With Independent Channels, or if you abort the session using NI-DCPower Abort With Channels.

NI-DCPower Reset With Channels and NI-DCPower Reset Device also place channels in the Uncommitted state.

Although you can configure properties in the Uncommitted state, NI-DCPower does not apply the properties until the Committed or Running states. For example, NI-DCPower Export Attribute Configuration verifies that the properties that you configure for a session are valid, but do not cause a transition to the Committed state. Thus, in the Uncommitted state, a channel remains in the same configuration as the last time that you commit a session.

Committed

Call NI-DCPower Commit With Channels to verify all properties, apply a select group of settings to channels, and transition to the Committed state.

NI-DCPower applies the following properties upon entering the Committed state.

- Aperture Time
- Aperture Time Units
- Auto Zero
- Current Compensation Frequency
- Current Gain Bandwidth
- Current Pole Zero Ratio
- Measure When
- Measure Record Length
- Measure Record Length Is Finite
- Merged Channels
- Output Capacitance
- Output Connected
- Output Resistance
- Power Line Frequency
- Power Source

- Pulse Bias Current Level
- Pulse Bias Current Limit
- Pulse Bias Current Limit High
- Pulse Bias Current Limit Low
- Pulse Bias Voltage Level
- Pulse Bias Voltage Limit
- Pulse Bias Voltage Limit High
- Pulse Bias Voltage Limit Low
- Reset Average Before Measurement
- Samples to Average
- Sense
- Sequence Loop Count Is Finite
- Sequence Loop Count
- Source Delay
- Source Mode
- Transient Response
- Voltage Compensation Frequency
- Voltage Gain Bandwidth
- Voltage Pole Zero Ratio
- All trigger-, event-, and routing-related properties

Note NI-DCPower applies any properties not listed in the table above to channels in the Running state.

If you configure a property on a channel multiple times, NI-DCPower applies only the most recent configuration upon entering the Committed state. If you modify any properties in the Committed state, the channel implicitly transitions back to the Uncommitted state, and the channel configuration continues to reflect the previously committed properties.

Call NI-DCPower Initiate With Channels from the Committed state to transition to the Running state.

Running

To place channels in the Running state, call NI-DCPower Initiate With Channels. In the Running state, a channel begins output signal generation and can acquire

measurements.

Note If you open a session with the NI-DCPower Initialize with Independent Channels, the outputs for the channels you specify with NI-DCPower Initiate With Channels to transition to the Running state are enabled, unless you explicitly disable the outputs of any channel(s) before initiating.

In the Running state, the channel configuration determines the behavior of the channel. For example, a channel can either configure a single output point or step through a sequence of points. Additionally, a channel can be configured to wait for triggers before performing an operation.

While running in Single Point source mode some properties and attributes can be dynamically reconfigured. Properties and attributes reconfigured dynamically (in the Running state) are immediately applied.

Note While running in sequence source mode, you can disable channel output after the Sequence Engine Done Event. No other properties are dynamically configurable while running a sequence.

In the Running state, there are several functions that you can use to take measurements, fetch buffered measurements, query the output state, or query the device state. You can call the only the following functions in the Running state.

- NI-DCPower Measure
- NI-DCPower Measure Multiple
- NI-DCPower Fetch Multiple
- NI-DCPower Send Software Edge Trigger With Channels
- NI-DCPower Query In Compliance
- NI-DCPower Query Output State
- NI-DCPower Wait For Event With Channels

Tip Call the NI-DCPower Abort With Channels function to transition from the Running state to the Uncommitted state. If Running in Sequence source mode, aborting cancels a sequence even if it has not yet run to completion.

Differences Between Sessions Created With Independent Channels and Deprecated Initialize Functions

The NI-DCPower Initialize With Independent Channels function replaces all prior NI-DCPower Initialize functions. NI recommends using Initialize With Independent Channels for new programs. Initializing with independent channels allows you to configure multiple channels of the same instrument, or of multiple instruments, independently of one another.

Sessions initialized with independent channels have fewer limitations than sessions initialized with other initialize functions.

Table 1. Behavior Changes for Initialize (Deprecated) and Initialize with Independent Channels

Feature	Behavior in Session Initialized With	
	Initialize With Independent Channels	**Any Other NI-DCPower Initialize Function**
Triggers and events	Configurable per channel	Not available in sessions that include multiple channels
Measure When property	Configurable per channel	Must be set to **On Demand** if a session includes multiple channels
Measure records	Available in sessions that include multiple channels	Not available in sessions that include multiple channels
Sequence source mode	Configurable per channel	Not available in sessions that include multiple channels

Note You cannot mix both simple and advanced sequences in the same Initialize With Independent Channels session.

Fully Qualified Channel Names and Initialize With Independent Channels

NI-DCPower Initialize With Independent Channels allows you to include channels from multiple physical instruments and configure them independently of one another. If the session spans multiple physical instruments, you must use a ***fully qualified channel name*** or a ***fully qualified channel range*** to unambiguously identify and configure the channels.

- **Fully qualified channel name**—An instrument name plus a channel. Example: `PXI1Slot3/0`
- **Fully qualified channel range**—An instrument name plus a channel range. Example: `PXI1Slot3/0-23` or `PXI1Slot3/0:23`

> **Note** If the session includes channels from a single physical instrument, you can use unqualified channel names (for example, `0-23` or `0:23`).

Related concepts:

- Single Point Source Mode
- Dynamic Reconfiguration
- Sequence Source Mode
- Advanced Sequencing

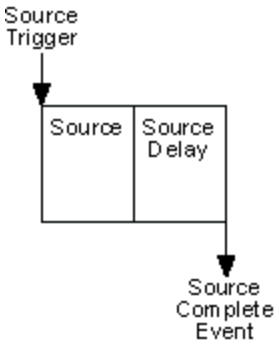
Single Point Source Mode

In Single Point source mode, the source unit applies a single source configuration when it enters the Running state. You can then update the source configuration dynamically.

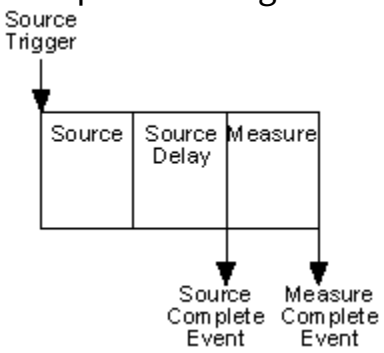
Single Point Source Model

The following steps illustrate the typical operation of the source unit when you initiate in Single Point source mode:

1. (Optional) The source unit waits for a Source trigger.
2. The source unit updates the source configuration.
3. After waiting the time specified by the NI-DCPower Source Delay property, the source unit generates a Source Complete Event, as illustrated in the following figure.



4. (Optional) If the NI-DCPower Measure When property is set to **Automatically After Source Complete**, the measure unit takes a measurement after the Source Complete event generates.



Note In Single Point source mode, only the Source trigger is available to control the source unit. Accordingly, the Sequence Advance trigger is not available, and Start trigger must be set to **None**.

Note The PXI-4110 and PXI-4130 do not support triggers and events or automatically measuring after source complete. On these devices, after entering the Running state, the source unit simply updates the source configuration without waiting for any triggers or generating any events, and the measure unit cannot automatically take a measurement after the source completes.

Related concepts:

- Programming States
- Dynamic Reconfiguration
- Sequence Source Mode

Dynamic Reconfiguration

While in the Running state, you can edit the source configuration dynamically using any VI, function, property or attribute listed in the tables below. When editing the source configuration dynamically, the changes are immediately applied. For sessions that include multiple channels and were created with NI-DCPower Initialize With Independent Channels, changes are applied to the channels in parallel.

The following output functions support dynamic reconfiguration:

- DC Voltage
- DC Current

Note Some VIs, functions, properties, and attributes are not supported on specific devices.

VI	Function
niDCPower Configure Voltage Level	`niDCPower_ConfigureVoltageLevel`
niDCPower Configure Voltage Level Range	`niDCPower_ConfigureVoltageLevelRange`
niDCPower Configure Voltage Limit	`niDCPower_ConfigureVoltageLimit`
niDCPower Configure Voltage Limit Range	`niDCPower_ConfigureVoltageLimitRange`
niDCPower Configure Current Level	`niDCPower_ConfigureCurrentLevel`
niDCPower Configure Current Level Range	`niDCPower_ConfigureCurrentLevelRange`
niDCPower Configure Current Limit	`niDCPower_ConfigureCurrentLimit`
niDCPower Configure Current Limit Range	`niDCPower_ConfigureCurrentLimitRange`
niDCPower Configure Output Enabled	`niDCPower_ConfigureOutputEnabled`

Property	Attribute
Voltage Level	`NIDCPOWER_ATTR_VOLTAGE_LEVEL`
Voltage Level Range	`NIDCPOWER_ATTR_VOLTAGE_LEVEL_RANGE`
Current Level	`NIDCPOWER_ATTR_CURRENT_LEVEL`

Property	Attribute
Current Level Range	NIDCPOWER_ATTR_CURRENT_LEVEL_RANGE
Voltage Limit	NIDCPOWER_ATTR_VOLTAGE_LIMIT
Voltage Limit High	NIDCPOWER_ATTR_VOLTAGE_LIMIT_HIGH
Voltage Limit Low	NIDCPOWER_ATTR_VOLTAGE_LIMIT_LOW
Voltage Limit Range	NIDCPOWER_ATTR_VOLTAGE_LIMIT_RANGE
Current Limit	NIDCPOWER_ATTR_CURRENT_LIMIT
Current Limit High	NIDCPOWER_ATTR_CURRENT_LIMIT_HIGH
Current Limit Low	NIDCPOWER_ATTR_CURRENT_LIMIT_LOW
Current Limit Range	NIDCPOWER_ATTR_CURRENT_LIMIT_RANGE
Output Enabled	NIDCPOWER_ATTR_OUTPUT_ENABLED
Output Connected	NIDCPOWER_ATTR_OUTPUT_CONNECTED
Output Resistance	NIDCPOWER_ATTR_OUTPUT_RESISTANCE
OVP Enabled	NIDCPOWER_ATTR_OVP_ENABLED
Self Calibration Persistence	NIDCPOWER_ATTR_SELF_CALIBRATION_PERSISTENCE
Source Delay	NIDCPOWER_ATTR_SOURCE_DELAY
Requested Power Allocation	NIDCPOWER_ATTR_REQUESTED_POWER_ALLOCATION

Dynamic reconfiguration is only available when Source trigger is disabled. Therefore, in Single Point source mode, the device can wait for and accept only one Source trigger and apply a single source configuration. If you want the device to wait for another Source trigger before reconfiguring the source, either use Sequence source mode or call NI-DCPower Abort With Channels followed by `Initiate With Channels` to start a new Single Point generation that waits for a trigger.

Related concepts:

- Programming States

Related information:

- NI-DCPower LabVIEW Reference

Sequence Source Mode

A ***sequence*** is composed of steps that specify a series of outputs for an NI-DCPower channel. In a ***simple sequence***, you can specify either a series of voltage outputs or a series of current outputs. ***Advanced sequencing*** allows you to configure multiple properties per sequence step.

For more information on using and customizing advanced sequences, refer to ***Advanced Sequencing***.

During a sequence, the channel steps through a predetermined set of NI-DCPower configurations without any interaction with the host system and NI-DCPower. Because the host system is not involved in executing the NI-DCPower sequence changes, the changes from one step in the sequence to the next are deterministic.

Support For Sequence Mode

- The following devices do not support this feature: PXI-4110, PXI-4130
- Using NI-DCPower Set Sequence and Advanced Sequence functions for the same channel in the same session is not supported.

Typical Sequence Operation

The following steps illustrate the typical operation of the source unit when you initiate in Sequence source mode.

1. (Optional: Advanced sequencing only) If the channel was previously in the Uncommitted state and you created a Commit step, the driver applies the Commit step to the channel.
   a. The source unit applies the voltage or current specified by the Commit step.
   b. The source unit waits for the source delay specified by the Commit step.

   Note The Commit step in an advanced sequence does not generate a Source Complete event.

2. (Optional) The source unit waits for a Start Trigger.
3. The channel steps through the sequence.

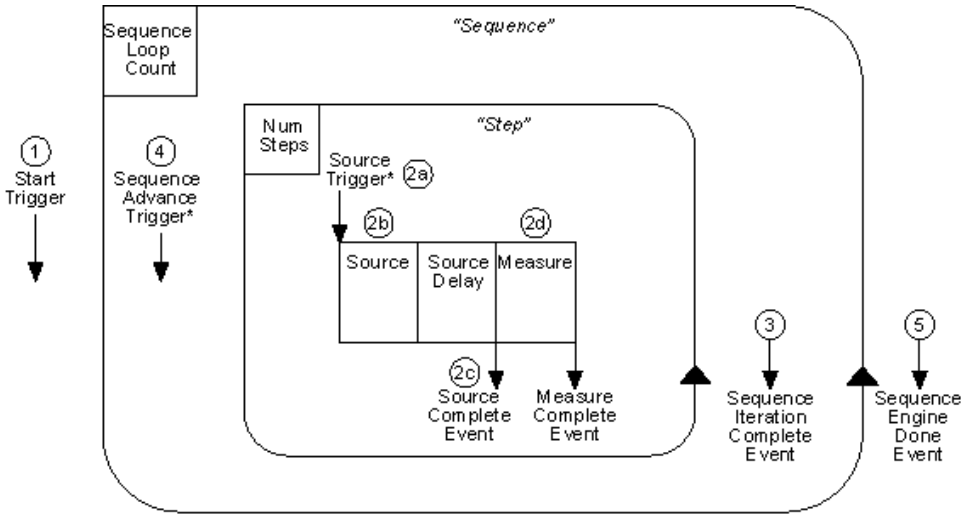
a. (Optional) If this is not the first step in the sequence, the source unit waits for a Source trigger.
b. The source unit applies the voltage or current in the next sequence step.
c. After waiting the time specified by the next source delay in the sequence, the source unit generates a Source Complete event.
d. (Optional) The measure unit takes a measurement and stores it in a buffer on the instrument.

> **Note** The presence of this step is dependent on the settings of NI-DCPower Measure When. If the Measure When property is set to `Automatically After Source Complete`, the measure unit takes a measurement after each source step. Otherwise, the measure unit operates independently of the source unit.

e. The source unit repeats 3a through 3d for each step in the sequence.

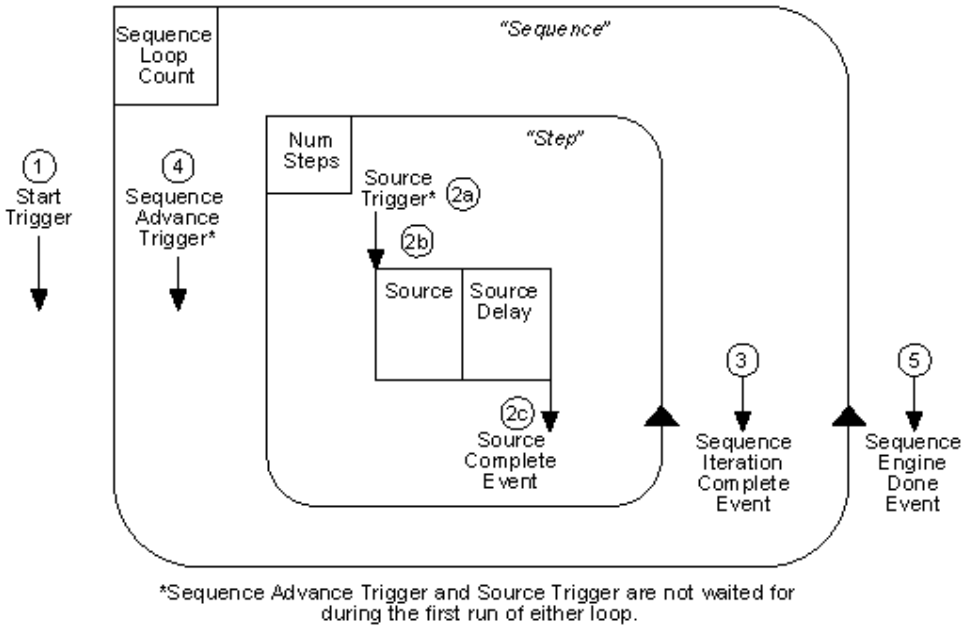
4. The sequence iteration is complete, and the source unit generates the Sequence Iteration Complete event.
5. (Optional) The source unit loops back and repeats the same sequence. When it loops back, the source unit waits for a Sequence Advance trigger (if it is not disabled) and then runs through the sequence again without waiting for the Start trigger.
6. If NI-DCPower Sequence Loop Count Is Finite is set to `True`, once the sequence runs the number of times specified by NI-DCPower Sequence Loop Count property, the operation is complete, and NI-DCPower generates the Sequence Engine Done event, as shown in the following figures. If NI-DCPower Sequence Loop Count Is Finite is set to `False`, the NI-DCPower Sequence Loop Count property is ignored, and the sequence returns to step 2a until it is aborted by calling the NI-DCPower Abort With Channels or by closing the session.

The following figure illustrates a sequence under the following conditions:



- NI-DCPower Measure When is set to `Automatically After Source Complete`.
- The source unit is not pulsing.

The following figure illustrates a sequence under the following conditions:



- NI-DCPower Measure When is not set to `Automatically After Source Complete`.
- The source unit is not pulsing.

Related concepts:

- Advanced Sequencing
- Single Point Source Mode
- Pulsing

Advanced Sequencing

Use advanced sequencing instead of the NI-DCPower Set Sequence function when you want more options for configuring a channel differently between sequence steps.

In each advanced sequence step, you can specify the state of an individual channel by configuring a large selection of properties.

Note For more information on simple sequencing, refer to NI-DCPower Set Sequence.

Differences Between Advanced Sequencing and Simple Sequencing

- Simple sequencing is configured using NI-DCPower Set Sequence. Advanced sequencing is configured using a combination of NI-DCPower properties, the NI-DCPower Create Advanced Sequence With Channels function, and the Create Advanced Sequence Step With Channels function.
- In an advanced sequence, you can change a greater selection of properties in each step than you can in a simple sequence.
  - Simple sequencing configures a series of voltage or current outputs and, optionally, corresponding source delays. Simple sequences are configured using Set Sequence.
  - Advanced sequencing allows you to choose which properties to change between sequence steps. Advanced sequences are configured using a combination of NI-DCPower properties, the Create Advanced Sequence function, and the Create Advanced Sequence Step With Channels function. For example, in an advanced sequence, you could also change the measurement aperture time for each step, or switch between voltage and current output for each step.
- In an advanced sequence, you can create a Commit step to configure channels to a known state before the sequence runs.

- Programming a device with advanced sequencing requires you to use specific advanced sequencing functions that are not used in simple sequencing.
- You can use the Export Attribute Configuration and Import Attribute Configuration functions to transfer advanced sequences between sessions, but not simple sequences.

Refer to the `NI-DCPower Advanced Sequence Changing Aperture Time.vi` or `NI-DCPower Advanced Sequence Changing Output Function.vi` examples in the LabVIEW Example Finder. For an example of simple sequencing using the niDCPower Set Sequence VI, refer to the `NI-DCPower Hardware-Timed Voltage Sweep.vi` example.

Support for Advanced Sequencing

The following devices do not support advanced sequencing:

- PXI-4110
- PXIe-4112/4113
- PXI-4130
- PXI-4132
- PXIe-4154

Note Using both simple sequencing and advanced sequencing for the same channel within the same session is not supported.

Related concepts:

- Sequence Source Mode

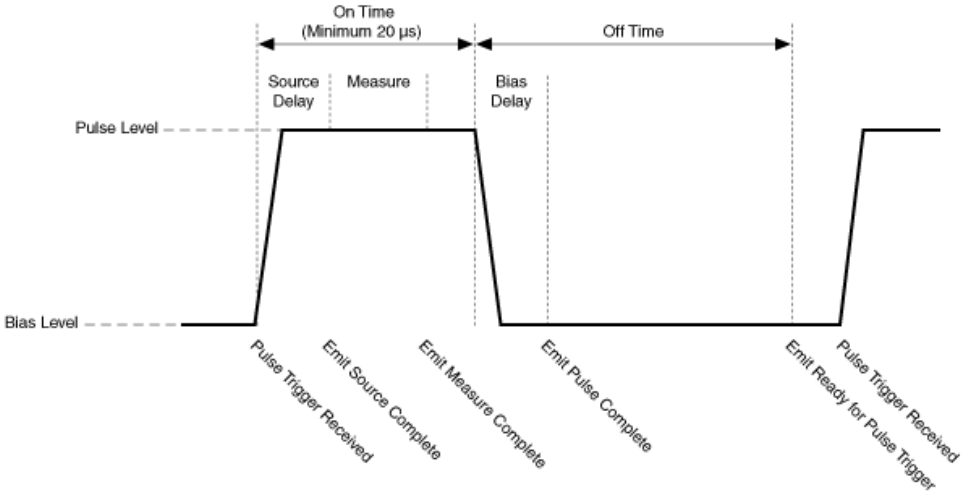
Pulsing

The source unit can output configurable voltage pulses or current pulses.

Pulsing in Single Point Source Mode

In Single Point source mode, the source unit generates a single pulse, then returns to the pulse bias level.

Figure 2. Single Pulse Cycle



The following steps illustrate the typical operation of the source unit for each pulse when you initiate a pulse output function.

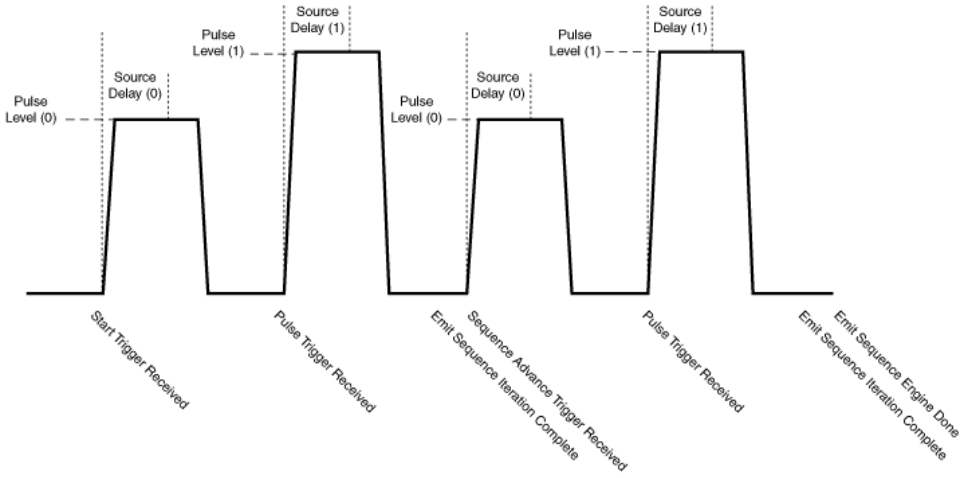
1. To enable pulsing, set NI-DCPower Output Function to **Pulse Voltage** or **Pulse Current**.
2. (Optional) In Sequence source mode, if this is not the first step in a sequence, the source unit emits a Ready For Pulse Trigger event and waits for a Pulse trigger.
3. After receiving the Pulse Trigger, the source unit applies the pulse level configuration. In Sequence mode, this is the next pulse level in the array.
4. After waiting the time specified by the source delay, the source unit generates a Source Complete event. In Sequence mode, if you specify source delay on a per-step basis then this time is optionally the next source delay in the array.
5. (Optional) To specify that the measure unit now takes a measurement and stores it in a buffer on the device, set NI-DCPower Measure When to **Automatically After Source Complete**.
6. After waiting the remainder of the pulse on time, the source unit applies the pulse bias configuration.
7. After waiting the pulse bias delay, the source unit emits a Pulse Complete event.
8. The source unit waits the remainder of the pulse off time.
9. At commit, the bias settings are applied to the output.

Pulsing in Sequence Source Mode

In Sequence source mode, the source unit applies a list of pulse levels in succession while still maintaining the specified duty cycle by applying the specified **Pulse On** and

Pulse Off times. The following figure illustrates a sequence consisting of two pulses. The sequence is repeated twice.

Figure 3. Sequence with Two Pulses



Note Timing properties for pulses, such as **pulse on** time, **pulse off time**, and **pulse bias delay**, must be constant for all steps in the sequence.

Pulsing in an Advanced Sequence

Pulsing in an advanced sequence differs from pulsing in sequence source mode because you can reconfigure all pulse-related properties per step in an advanced sequence.

Pulse Source Model

Table 2. Properties Applied for Each Pulse Output Function

Pulse Voltage	Pulse Current
Pulse Voltage Level	Pulse Current Level
Pulse Bias Voltage Level	Pulse Bias Current Level
Pulse Voltage Level Range	Pulse Current Level Range
Pulse Current Limit[1]	Pulse Voltage Limit[1]
Pulse Current Limit High[2]	Pulse Voltage Limit High[2]
Pulse Current Limit Low[2]	Pulse Voltage Limit Low[2]

Pulse Voltage	Pulse Current
Pulse Bias Current Limit[1]	Pulse Bias Voltage Limit[1]
Pulse Bias Current Limit High[2]	Pulse Bias Voltage Limit High[2]
Pulse Bias Current Limit Low[2]	Pulse Bias Voltage Limit Low[2]
Pulse Current Limit Range	Pulse Voltage Limit Range

[1]Applied when NI-DCPower Compliance Limit Symmetry is set to **Symmetric**.

[2]Applied when NI-DCPower Compliance Limit Symmetry is set to **Asymmetric**.

The following properties are applied for both pulse output functions:

- Source Delay
- Pulse Bias Delay
- Pulse On Time
- Pulse Off Time

Support for Pulsing

The following devices do not support pulsing.

- PXI-4110
- PXIe-4112/4113
- PXI-4130
- PXI-4132
- PXIe-4140/4141/4142/4143/4144/4145
- PXIe-4154
- PXIe-4162/4163

Related concepts:

- Sequence Source Mode
- Advanced Sequencing

Detecting Internal/Auxiliary Power

Use an auxiliary power source to enable the full-power capabilities of NI instruments that support this feature. NI-DCPower can dynamically query whether the auxiliary power is connected to the device.

The NI-DCPower Auxiliary Power Source Available property returns TRUE if auxiliary power is connected to the device and FALSE if only internal power is available on the device.

NI instruments cannot differentiate between an auxiliary power loss or a blown fuse on the auxiliary power input line. If auxiliary power is properly connected and Auxiliary Power Source Available returns FALSE, you might need to replace the auxiliary power input fuse.

Simulating an Instrument

Simulate an instrument using NI-DCPower or Measurement & Automation Explorer (MAX) to develop, modify, and/or test an application without hardware.

Using a simulated device to test an application eliminates the risk of hardware damage. Additionally, you can use a simulated instrument to evaluate an NI product for which you do not have hardware.

Tip As with any installed and configured instrument, you can use InstrumentStudio to test the basic functionality of the device.

Related concepts:

- Using InstrumentStudio with Your NI-DCPower Instrument

Simulating an Instrument with NI-DCPower

Complete the following steps to create and configure a simulated instrument using NI-DCPower.

1. Run NI-DCPower Initialize With Independent Channels.

2. Regardless of whether you want to simulate a single-instrument or multi-instrument session, set both the `resource name` and `option string` parameters. The `option string` parameter is composed of the `Simulate` and `Driver Setup` keywords, as illustrated in the following example.
For a single-instrument session, use the following:

Parameter	Syntax

resource name	For single-instrument sessions:
	
	`<resource name>`
	
	For multi-instrument sessions:
	
	`<resource name 1>, <resource name 2>`
	
	Where resource name is either an instrument name (`PXI1Slot2`), a fully qualified channel name (`PXI1Slot2/0`), or a fully qualified channel range (`PXI1Slot2/0-3`).

option string	For single-instrument sessions:
	
	`Simulate=1,DriverSetup=Model:<model number>;BoardType:<bus connector>`
	
	For multi-instrument sessions:
	
	`Simulate=1,DriverSetup=ResourceName:<resource name 1>;Model:<model number>;BoardType:<bus connector>&ResourceName:<resource name 2>;Model:<model number>;BoardType:<bus connector>`

Note The `ResourceName` entry of the option string may contain a single instrument names only (`PXI1Slot2`) and may contain only a single instrument name per `ResourceName` entry.

Simulation Examples

To simulate a single PXI-4110, use the following `option string` syntax:
`Simulate=1,DriverSetup=Model:4110;BoardType:PXI`

You can also simulate a subset of channels for a single instrument using the `resource name` parameter of the Initialize With Independent Channels VI.

To simulate a multi-instrument session with a PXI-4110 and PXIe-4163, use the following syntax:

- `resource name`: `PXI1Slot2/0-2,PXI1Slot3/0-23`
- `option string`:
  `Simulate=1,DriverSetup=ResourceName:PXI1Slot2;Model:4110;BoardTy`

To simulate a multi-instrument session with these instruments that includes only a subset of their channels, use the following syntax:

- `resource name`: `PXI1Slot2/0,PXI1Slot3/0,PXI1Slot3/2,PXI1Slot3/4,PXI1Slot3/6,PXI1Slot3/8,PXI1Slot3/12`
- `option string`:
  `Simulate=1,DriverSetup=ResourceName:PXI1Slot2;Model:4110;BoardTy`

Only those channels that you explicitly include are simulated.

Driver Setup String

The `DriverSetup` keyword is set using the driver setup string, and comprises the resource name (multi-instrument sessions only), instrument model number, and bus connector.

When you specify the driver setup string, NI-DCPower uses the `option string` parameter of niDCPower Initialize With Independent Channels to determine the following:

- How many instruments to simulate;
- Which channel numbers to use; and
- Which channels belong on which simulated instrument.

If you do not specify the driver setup string, NI-DCPower simulates the device specified in the **resource name** parameter.

You do not have to specify a value for all the `option string` properties. If you do not specify a value for a property, NI-DCPower uses the default value. If you specify a `Model` but not a `BoardType`, the driver infers the board type. If you specify neither the driver setup string nor the `resource name` parameter, or if you do include a driver setup string that excludes a `Model`, NI-DCPower simulates the device by default.

Simulating an Instrument with MAX

Complete the following steps to create and to configure a simulated power supply in Measurement & Automation Explorer (MAX).

1. Launch MAX.
2. Right-click **Devices and Interfaces** in the MAX configuration tree, and select **Create New...**.
3. Select **Simulated NI-DAQmx Device or Modular Instrument**, then click **Finish**.
4. In the Create Simulated NI-DAQmx Device window, expand **Power Supplies**.
5. Select the power supply to simulate, then click **OK**.

The power supply appears in the MAX configuration tree with a yellow icon indicating that the device is simulated.

Import/Export Attribute Configuration Mapping Behavior

When importing and exporting configurations between NI-DCPower sessions that are initialized with different channels, the configurations of the exporting channels are mapped to the importing channels based on the order of the resources you specify in the **resource name** input to the Initialize With Independent Channels node for each session.

Note NI-DCPower returns an error if the total number of channels initialized for the exporting session is not equal to the total number of channels

> initialized for the importing session. However, as long as the number of initialized channels is equal between sessions, you can export and import configurations between any number of physical instruments. For each channel mapping created, NI-DCPower checks that the model of an exporting instrument channel matches the model of the importing instrument channel.

In general, NI-DCPower configurations are mapped between channels in the exporting and importing sessions as follows.

Table 3. NI-DCPower Configuration Mapping Between Channels

Case	Resource(s), Exporting Session	Resource(s), Importing Session	Configuration Exported From	Imported To
Single physical instrument per session	`PXI1Slot3/0-1`	`PXI1Slot4/1-2`	`PXI1Slot3/0`	`PXI1Slot4/1`
			`PXI1Slot3/1`	`PXI1Slot4/2`
>1 physical instrument in either session, equal numbers	`PXI1Slot4/3,PXI1Slot5/3`	`PXI1Slot2/0,PXI1Slot3/0`	`PXI1Slot4/3`	`PXI1Slot2/0`
			`PXI1Slot5/3`	`PXI1Slot3/0`
>1 physical instrument in either session, export to fewer	`PXI1Slot4/3,PXI1Slot5/3`	`PXI1Slot3/0-1`	`PXI1Slot4/3`	`PXI1Slot3/0`
			`PXI1Slot5/3`	`PXI1Slot3/1`
>1 physical instrument in either session, export to more	`PXI1Slot3/0-1`	`PXI1Slot4/3,PXI1Slot5/3`	`PXI1Slot3/0`	`PXI1Slot4/3`
			`PXI1Slot3/1`	`PXI1Slot5/3`

Instrument-based properties that are exported and imported between sessions are mapped in the same manner according to channel order at the instrument level rather than at the channel level.

Import/Export Mapping Exceptions

Some properties have unique import/export mapping behavior.

For the Merged Channels property, only the instrument name is remapped, not the channel name. For example, when importing the value `PXI1Slot2/0` to a session on `PXI1Slot3/1`, the configuration is remapped to `PXI1Slot3/0`.

Mapping Behavior for Fully Qualified Terminal Names

When you export and import configurations that include properties that accept string-type fully qualified terminal names, the terminal names are mapped based on additional criteria specific to terminal names.

These criteria include:

- Whether the Engine number you specify matches the number of its corresponding channel
- Whether you specify the Engine number at all
- Whether the instrument alias specified for a terminal matches an instrument alias belonging to the session

Unqualified terminal names are not mapped.

Fully qualified terminal names are mapped according to the following general patterns:

Table 4. Mapping Behavior for Fully Qualified Terminal Names

Case	Instrument Aliases, Exporting Session	Instrument Aliases, Importing Session	Terminal Name Value, Exporting Session	Mapped Terminal Name in Importing Session
Equal number of physical instruments per session	`Dev1/0,Dev2/0`	`Dev3/1,Dev4/1`	`/Dev1/PXI_Trig0`	`/Dev3/PXI_Trig0`
			`/Dev2/PXI_Trig3`	`/Dev4/PXI_Trig3`
			`/Dev1/Engine0/MeasureTrigger`	`/Dev3/Engine1/MeasureTrigger`
			`/Dev2/Engine0/MeasureTrigger`	`/Dev4/Engine1/MeasureTrigger`
			`/Dev1/Engine5/MeasureTrigger`	`/Dev3/Engine5/MeasureTrigger`
			`Engine0/MeasureTrigger`	`Engine0/MeasureTrigger`

Case	Instrument Aliases, Exporting Session	Instrument Aliases, Importing Session	Terminal Name Value, Exporting Session	Mapped Terminal Name in Importing Session
			/Dev5/Engine0/MeasureTrigger	/Dev5/Engine0/MeasureTrigger
>1 physical instrument in either session, export to fewer	Dev1/0,Dev2/0	Dev3/0-1	/Dev1/PXI_Trig0	/Dev3/PXI_Trig0
			/Dev2/PXI_Trig3	/Dev3/PXI_Trig3
			/Dev1/Engine0/MeasureTrigger	/Dev3/Engine0/MeasureTrigger
			/Dev2/Engine0/MeasureTrigger	/Dev3/Engine1/MeasureTrigger
			/Dev1/Engine5/MeasureTrigger	/Dev3/Engine5/MeasureTrigger
			Engine0/MeasureTrigger	Engine0/MeasureTrigger
			/Dev5/Engine0/MeasureTrigger	/Dev5/Engine0/MeasureTrigger
>1 physical instrument in either session, export to more	Dev1/0-1	Dev2/0,Dev3/0	/Dev1/PXI_Trig0	/Dev2/PXI_Trig0
			/Dev1/PXI_Trig3	/Dev2/PXI_Trig3
			/Dev1/Engine0/MeasureTrigger	/Dev2/Engine0/MeasureTrigger
			/Dev1/Engine1/MeasureTrigger	/Dev3/Engine0/MeasureTrigger
			/Dev1/Engine5/MeasureTrigger	/Dev2/Engine5/MeasureTrigger
			Engine0/MeasureTrigger	Engine0/MeasureTrigger
			/Dev5/Engine0/MeasureTrigger	/Dev5/Engine0/MeasureTrigger