Custom NI VeriStand Steps for NI TestStand

v2.0

Table of Contents

[Initializing and Stopping the VeriStand Engine 3](#_Toc521235310)

[Initialize VeriStand 3](#_Toc521235311)

[Stop VeriStand 3](#_Toc521235312)

[Executing Tests 3](#_Toc521235313)

[Channel Value Limit Test 3](#_Toc521235314)

[RT Sequence Numeric Limit Test 4](#_Toc521235315)

[RT Sequence Pass Fail Test 5](#_Toc521235316)

[Get Channel Value 6](#_Toc521235317)

[Set Channel Value 6](#_Toc521235318)

[Get Multiple Channel Values 7](#_Toc521235319)

[Set Multiple Channel Values 8](#_Toc521235320)

[Alarms 8](#_Toc521235321)

[Set Alarm State 8](#_Toc521235322)

[Get Alarm List 9](#_Toc521235323)

[Get Alarm State 9](#_Toc521235324)

[Get Channel Fault Value 10](#_Toc521235325)

[Set Channel Fault Value 11](#_Toc521235326)

[Clear Fault 11](#_Toc521235327)

[Clear All Faults 12](#_Toc521235328)

[Get Model Parameter Value 12](#_Toc521235329)

[Update Model Parameters From File 13](#_Toc521235330)

[Set Model Parameter Value 13](#_Toc521235331)

[RT Sequence Action 14](#_Toc521235332)

[Logging Test Data 15](#_Toc521235333)

[Start VS Logging 15](#_Toc521235334)

[Stop VS Logging 16](#_Toc521235335)

[Understanding References 17](#_Toc521235336)

[Edit-Time Variables 17](#_Toc521235337)

[Run-Time References 17](#_Toc521235338)

[Using a Subsequence in the Same Sequence File 18](#_Toc521235339)

[Edit-Time 18](#_Toc521235340)

[Run-Time 18](#_Toc521235341)

[Using a Subsequence in a Different Sequence File 18](#_Toc521235342)

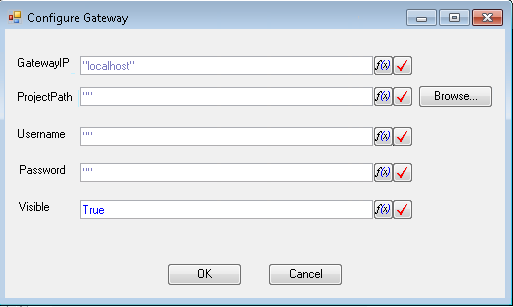
[Edit-Time 18](#_Toc521235343)

[Run-Time 18](#_Toc521235344)

# Initializing and Stopping the VeriStand Engine

## Initialize VeriStand

This step starts VeriStand, opens all the necessary references, and runs the configured project. It is responsible for creating the VeriStand container in FileGlobals. This variable contains the Gateway IP Address, the System Definition path, and the project path. It also contains the .NET references for the Fault Manager, Model Manager, Alarm Manager, Workspace, RT Sequence, Project, and Stimulus Profile. It is recommended that this step be placed in the Setup section of a sequence.



**GatewayIP – IP Address of the host gateway as a string.**

**ProjectPath – Path of the NI VeriStand project file as a TestStand string. You can use the Browse… button to browse to a path on disk.**

**Username – Username of the project.**

**Password – Password of the project.**

**Visible – If True the deployment windows appears when the project is deployed.**

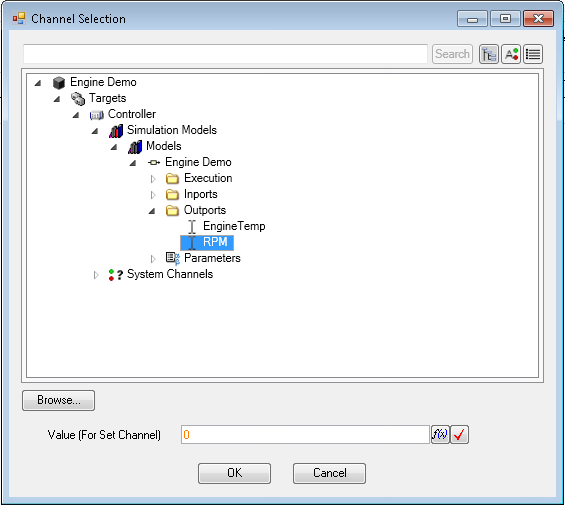
## **Stop VeriStand**

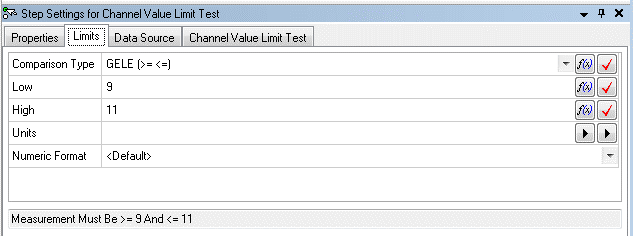
This step undeploys and closes the project defined in the VeriStand.ProjectPath File Global. It is recommended that this step be placed in the Clean section of a sequence.

# Executing Tests

## Channel Value Limit Test

This step gets the value of channel and determines if it is within the set limits. If the value is within the limits this step passes, otherwise it fails.





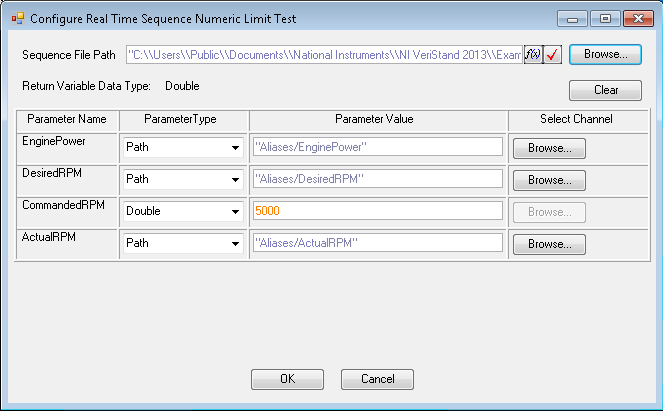
**Comparison Type – Determines which comparison to use from <=, >=, ==, !=, etc.**

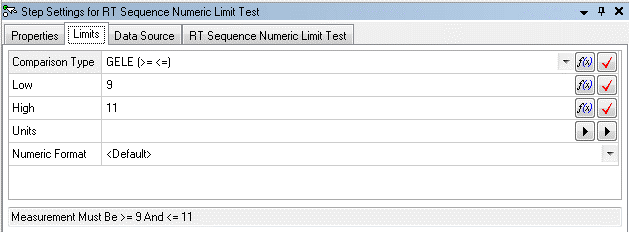
**Low – The low limit to use for the comparison**

**High – The high limit to use for the comparison**

## **RT Sequence Numeric Limit Test**

This step runs a real-time sequence and uses the numeric return variable to determine if it is within the expected limits. If the value is within the limits this step passes, otherwise it fails. The associated dialog pre-loads the parameters with default values but configuration is enabled through the Browse… button and editing the parameter value.





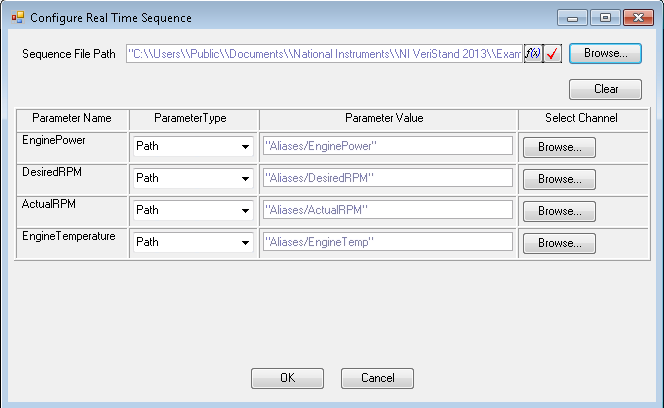
**Comparison Type – Determines which comparison to use from <=, >=, ==, !=, etc.**

**Low – The low limit to use for the comparison**

**High – The high limit to use for the comparison**

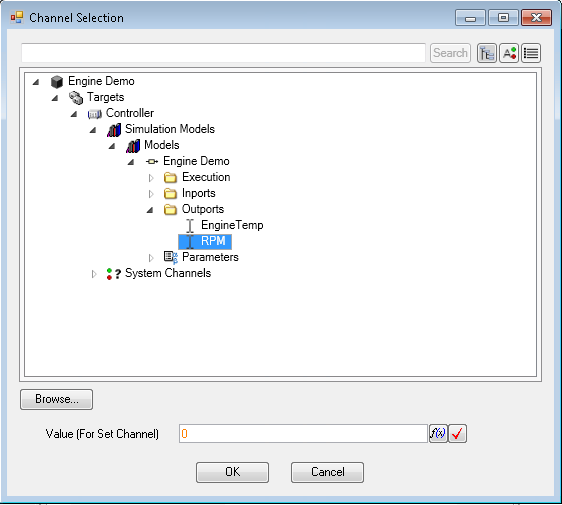
## **RT Sequence Pass Fail Test**

This step runs a real-time sequence and uses the Boolean return variable to determine if the test has passed or failed. The associated dialog pre-loads the parameters with default values but configuration is enabled through the Browse… button and editing the parameter value.



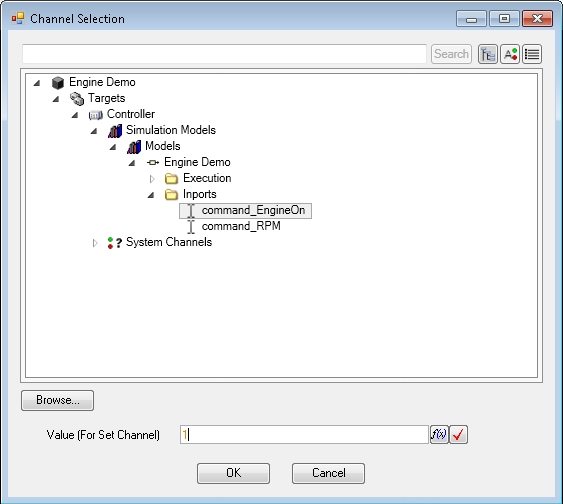
## Get Channel Value

This step gets a single channel value and places it in the Result.Numeric container.



## Set Channel Value

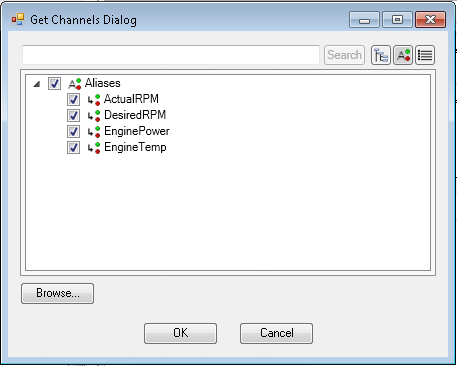
This step sets a single channel value using the Veristand.ValueToSet and Veristand.ChannelName step variables. The step blocks until the channel update has been confirmed.



**Value (For Set Channel) – Value that the channel will be set to**

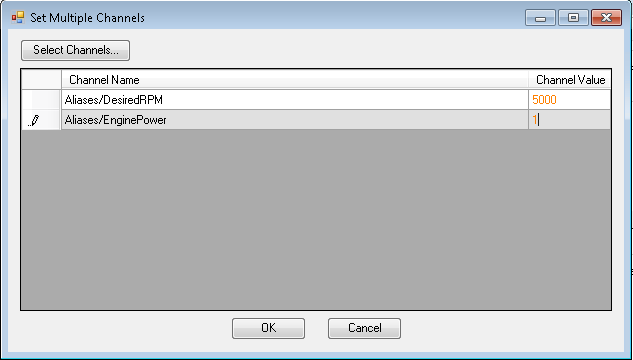
## **Get Multiple Channel Values**

This step gets multiple channel values. The channel names are placed in the Veristand.ChannelNames variable. The channel values are placed in the Veristand.ChannelValues variable.



## Set Multiple Channel Values

This step sets multiple channel values using the VeriStand.ValuesToSet and VeriStand.ChannelNames. The step blocks until the channel updates have been confirmed.



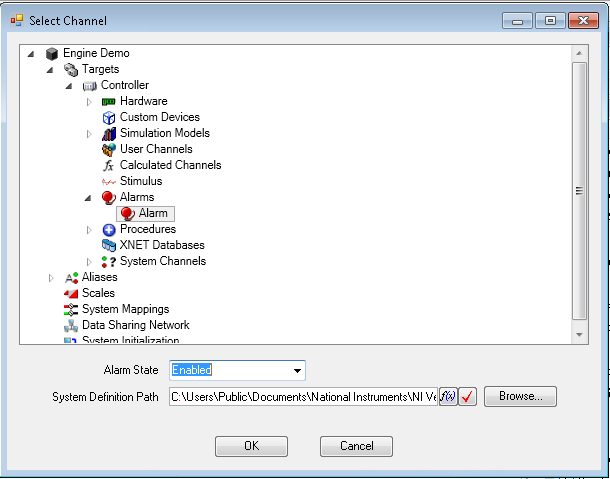
**Select Channels… - Loads a dialog to allow the user to select which channels to set**

**Channel Value – The value to set each channel to**

# **Alarms**

## Set Alarm State

This step sets an alarm to a desired state. The parameters are stored in the Veristand.FullAlarmPath, Veristand.AlarmState, and Veristand.AlarmName step variables.



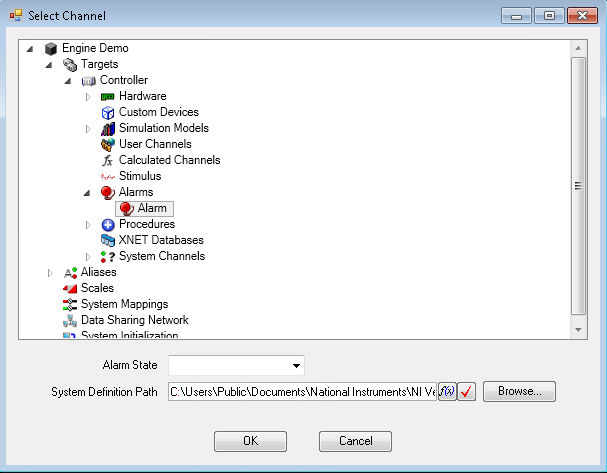
**Alarm State – The desired alarm state.**

## **Get Alarm List**

This step returns an array of strings containing all of the alarms in the system definition file. The value is stored in Result.Alarms.

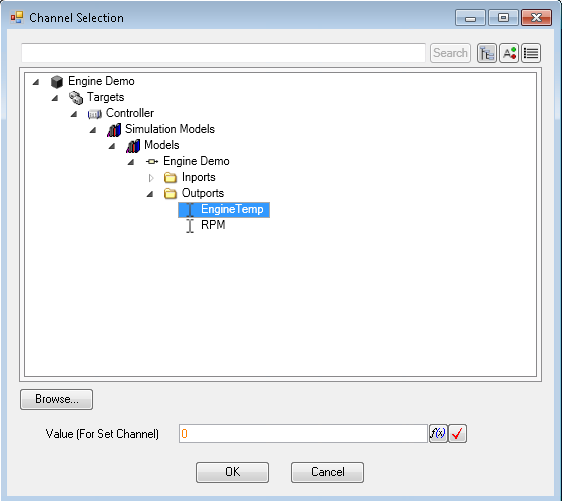
## Get Alarm State

This step gets an alarm state. The state is stored in the Result.Veristand.AlarmState variable.



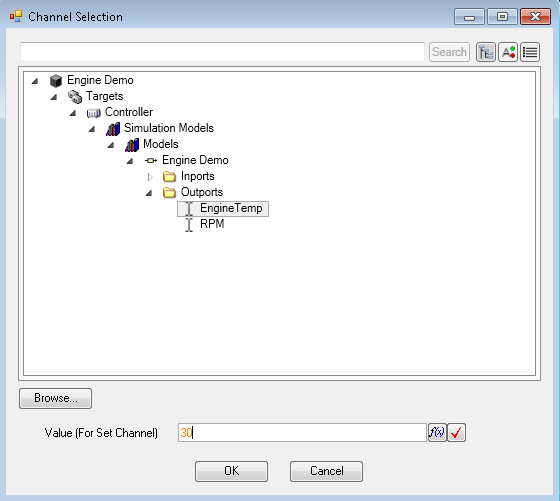
## Get Channel Fault Value

This step gets the fault value for the specified channel. The value is stored in the Result.Veristand.Numeric variable. The faulted state (True or False) is stored in Result.Veristand.Faulted. The channel name is stored in Result.Veristand.Channel.



## Set Channel Fault Value

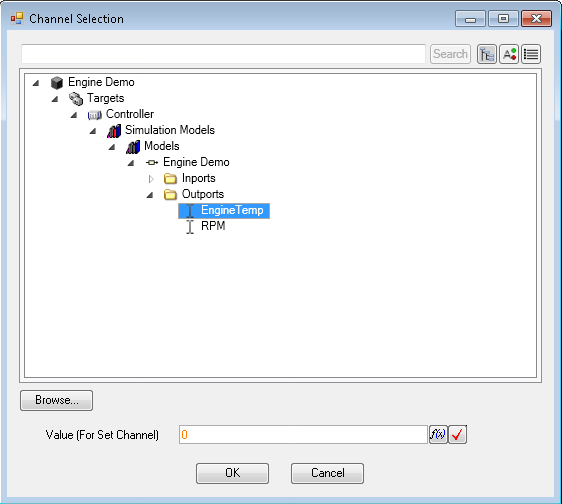
This step sets the fault value for the specified channel using the Veristand.ValueToSet and Veristand.ChannelName variables.



**Value (For Set Channel) – Value that the channel will be faulted to**

## Clear Fault

This step clears the fault on the specified channel.

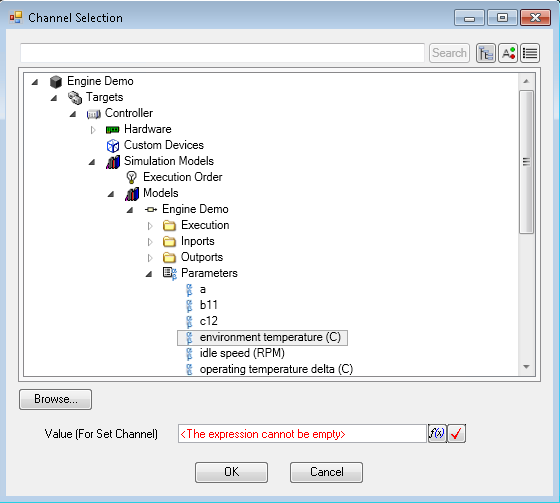


## Clear All Faults

This step clears all faults currently active on the deployed system.

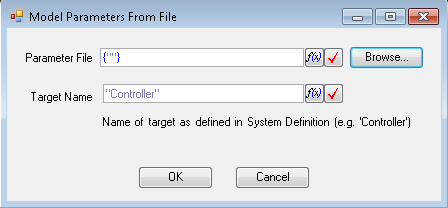
## Get Model Parameter Value

This steps gets the value of the specified model parameter and stores it in the Result.Veristand.ModelNumeric variable.



## Update Model Parameters From File

This step sets a number of model parameters at once using a specified \*.m model parameter file.

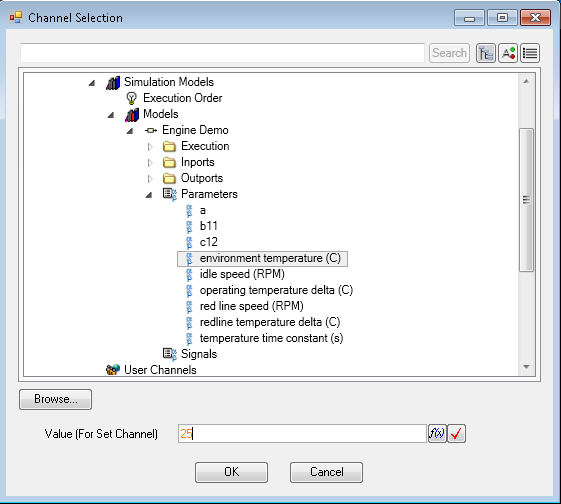


**Parameter File – The location of the parameter files as an array of strings**

**Target Name – The name of the target defined in the system definition file as a string**

## **Set Model Parameter Value**

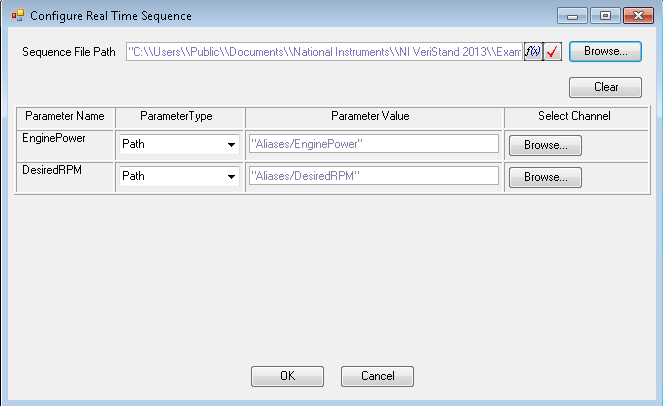
This step sets the value of a specified model parameter using the Veristand.FullChannelPath, Veristand.TargetName, Veristand.ValueToSet, and Veristand.ChannelName variables.



**Value (For Set Channel) – Value that the parameter will be set to**

## RT Sequence Action

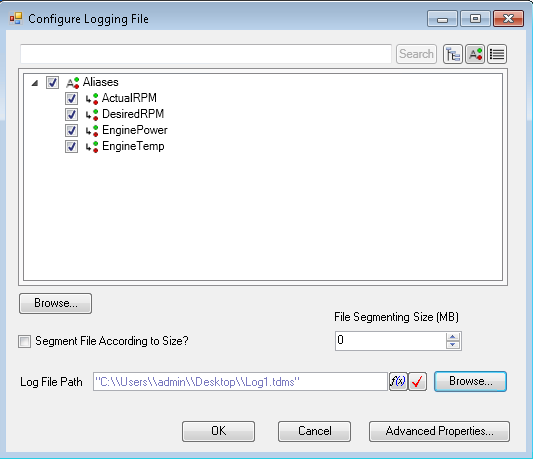
This step runs a real-time sequence. No evaluation of the return variable is performed. The associated dialog pre-loads the parameters with default values but configuration is enabled through the Browse… button and editing the parameter value.



# Logging Test Data

## Start VS Logging

This step begins logging on the host computer. The user has the ability to select which channels to log, choose whether or not to segment, set the log path, and add advanced properties for the TDMS file. This step must always be followed by a Stop VS Logging step in the sequence.

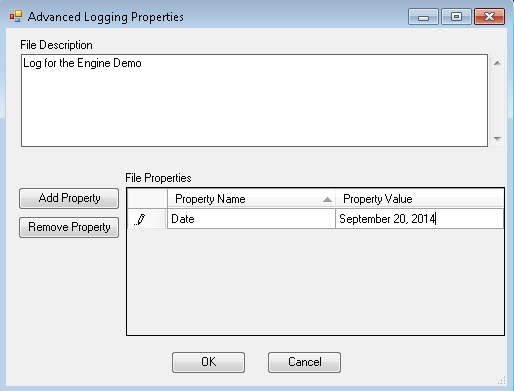


**Segment File According to Size? – True if the user would like to segment the file with size**

**File Segmenting Size (MB) – Size of the file to segment at if the Segment File According to Size? Is selected**

**Log File Path – Location of the generated log file on the host computer**

**Advanced Properties – Button to launch the Advanced Logging Properties dialog box**



**File Description – Optional box to add a description to the log file**

**Add Property – Adds a property to the File Properties table**

**Remove Property – Removes a property from the File Properties table**

**File Properties – The list of properties which will be written to the log file**

## **Stop VS Logging**

This step stops logging on the host computer.

# Understanding References

Example sequences to demonstrate the use of references are available in:

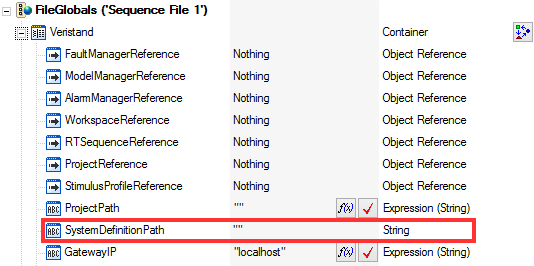
**<Public Documents>\National Instruments\NI VeriStand Steps for TestStand\Examples\References and Subsequences**

All references used by the step types are stored in one of two ways.

1. LabVIEW Packed Project Library containing run-time code modules (Run-Time Only)
   1. Project Reference
   2. Workspace Manager Reference
   3. Model Manager Reference
   4. Alarm Manager Reference
   5. Fault Manager Reference
   6. RT Sequence Reference
   7. Stimulus Profile Reference
2. TestStand File Globals (Run-Time)
   1. Gateway IP
   2. Project Path
3. TestStand File Globals (Edit-Time)
   1. System Definition Path

### Edit-Time Variables

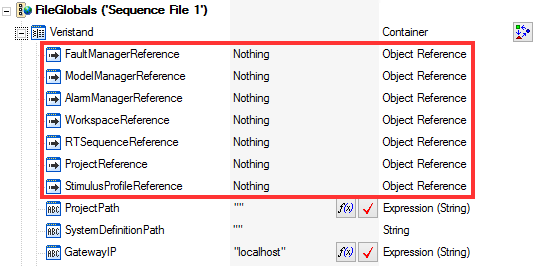
The only File Global variable required to configure the step types using the configuration GUIs is the *SystemDefinitionPath* File Global. This variable is created by both the *Initialize VeriStand* and *Open References* step types. This variable can also be created and populated manually. The variables must be populated manually if using the *Open References* step.



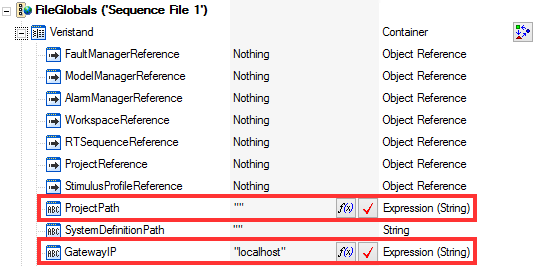
### Run-Time References

References used by the step types at run-time are stored within the *veristand-steps-runtime.lvlibp* library. These references are created at runtime by both the *Initialize VeriStand* and *Open References* step types. References are stored inside a LabVIEW functional global variable (FGV) within the packed project library.

The *Initialize VeriStand* and *Open References* step types also populate the same references (Project Reference, Workspace Reference, etc.) within the sequence File Globals. These references are not used by the run-time code modules. These references stored as File Globals can be used to interface with the running VeriStand project using other custom step types or using TestStand’s .NET adapter to call VeriStand’s .NET API directly.



The only File Globals **required** at run-time are the *GatewayIP* and *ProjectPath*. The GatewayIP is used by several steps including all RT Sequence step types. The *ProjectPath* variable is only used by the *Initialize VeriStand* or advanced project control step types.



It is possible to use the VeriStand Step Types in subsequences. Subsequences can be located either within the same sequence file as the calling sequence or within a separate sequence file.

## Using a Subsequence in the Same Sequence File

### Edit-Time

It is not necessary to create or pass any references or variables when configuring a subsequence located in the same sequence file as the calling sequence. The subsequence will use the *SystemDefinitionPath* variable located in the parent sequence file.

### Run-Time

It is not necessary to create or pass any reference or variables when executing a subsequence located in the same sequence file as the calling sequence. All run-time references are stored within the LabVIEW PPL and will be accessible to any sequence which accesses this library. This still applies if running the subsequence in a new thread or execution.

## Using a Subsequence in a Different Sequence File

### Edit-Time

The *SystemDefinitionPath* File Global must be present and populated to configure the step types in a separate sequence file. This File Global can be created manually or by using the *Open References* step type.

### Run-Time

The *GatewayIP* and *ProjectPath* File Globals are required by some step types. These must be passed to the child sequence if the sequence is in a separate file. The variables can be passed from the parent sequence to the child sequence using Parameters. It’s also possible to set the File Global values in the child sequence using the TestStand API. This still applies if running the subsequence in a new thread or execution.