

Issue date: 2017-04-04

Question: How to pass data from a Teststand sequence to the FlexStand GUI?

Applies to: All FlexStand OI versions

Answer:

UIMessage

It is often desired to pass measured data from a TestStand sequence to the plugin in FlexStand. For simple parameters like numerics, Booleans and strings these can be send using the Teststand UIMsg system.

For other data types, like for example frequency spectrums or other complex data, VI server can be used to transfer data.

This FAQ describes a method that uses VI server technology.

Background

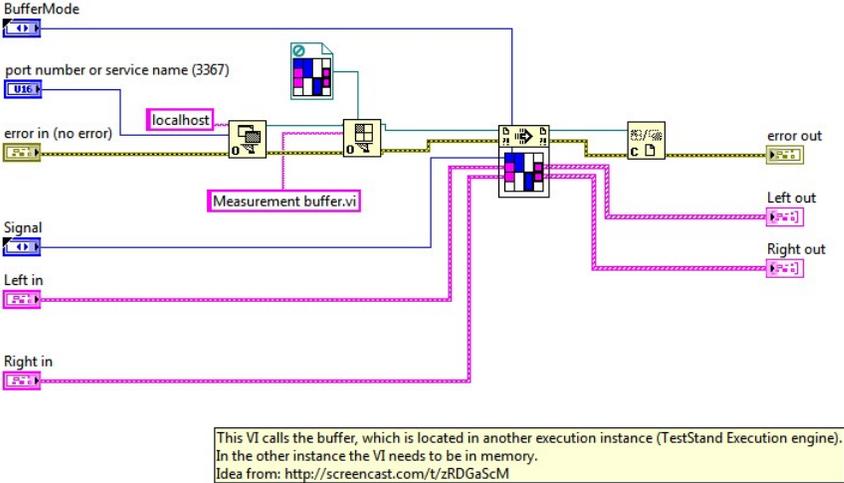
The problem of transferring data occurs because the LabVIEW instance that TestStand uses for executing the sequence is different from the LabVIEW instance that FlexStand uses for the GUI.

This means that the FlexStand GUI cannot just call a VI used by the steptypes because these run in a different instance.

By using the method of remote VI server call, an instance of LabVIEW can call any other VI on any other PC, if access is not blocked by firewalls.

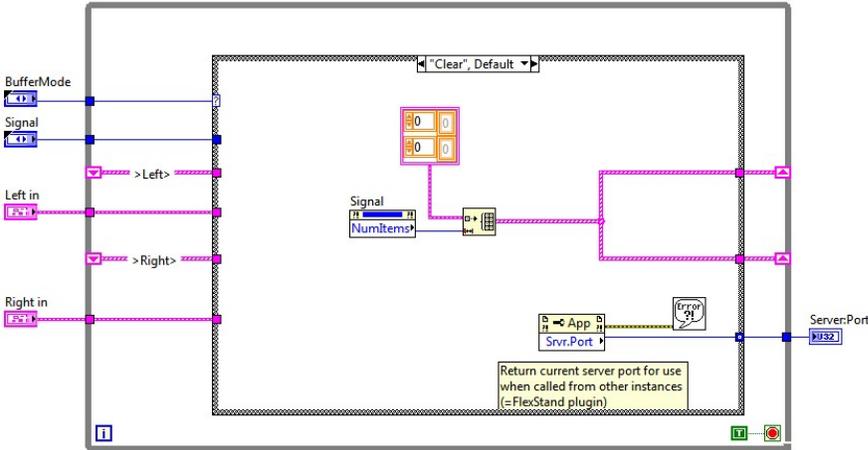
FLEXSTAND Operator Interface

Frequently Asked Question



Example

In the following example, we assume that the captured data are stored in a functional global. This functional global is used by the steps in the Teststand sequence for storing data.



This functional global stores a number of frequency spectras for left and right channel. This is an example and the functional global can hold any data types. The functional global has modes of Clear, Write and Read.

Note that this functional global also returns the server port used by this instance of LabVIEW. This port number must be used by the remote server to access this VI when



FLEXSTAND Operator Interface

Frequently Asked Question

reading the data (mode=read). All other code in this functional global is as a standard functional global.

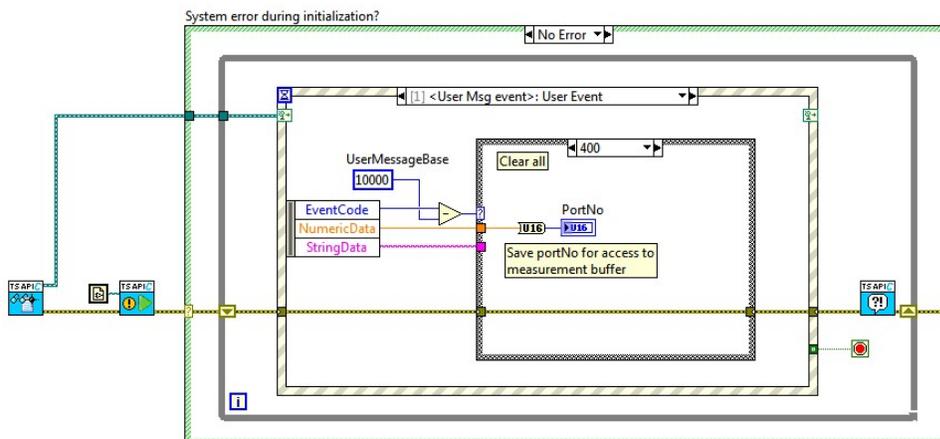
The server port number can now be send to the FlexStand GUI plugin using a standard UIMsg:

The screenshot shows the LabVIEW Step Settings dialog for the step 'Init Meas. graphs on GUI'. The 'Call Method' is set to 'PostUIMessageEx'. The 'Value' field is set to 'UIMsg_UserMessageBase+400' and 'Locals.ServerPort', which is highlighted with a red box. The 'Parameter Name' table below is as follows:

Parameter Name	Type	Direction	Log	Default	Value
eventCode	UIMessageCodes (Number (Signed 32...	in	<input type="checkbox"/>		UIMsg_UserMessageBase+400
numericDataParam	Number (64-bit Floating Point)	in	<input type="checkbox"/>		Locals.ServerPort
stringDataParam	String	in	<input type="checkbox"/>		''
activeXDataParam	Object Reference	in	<input type="checkbox"/>		''
synchronous	Boolean	in	<input type="checkbox"/>		False

In this example the ServerPort is stored in a local variable.

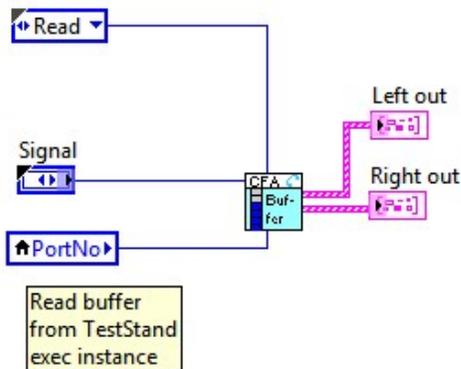
The FlexStand GUI plugin receives the UIMsg and stores the port number for later use.



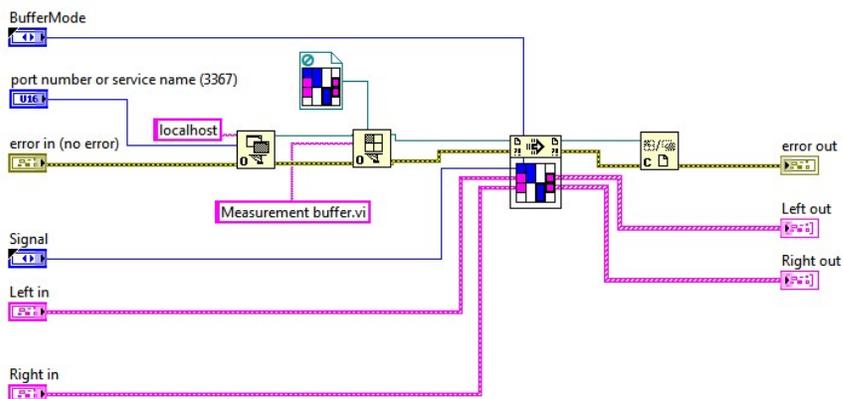
FLEXSTAND Operator Interface

Frequently Asked Question

When the FlexStand GUI should update the data on the GUI, it calls a VI that, via VI server, executes the VI in the other instance:



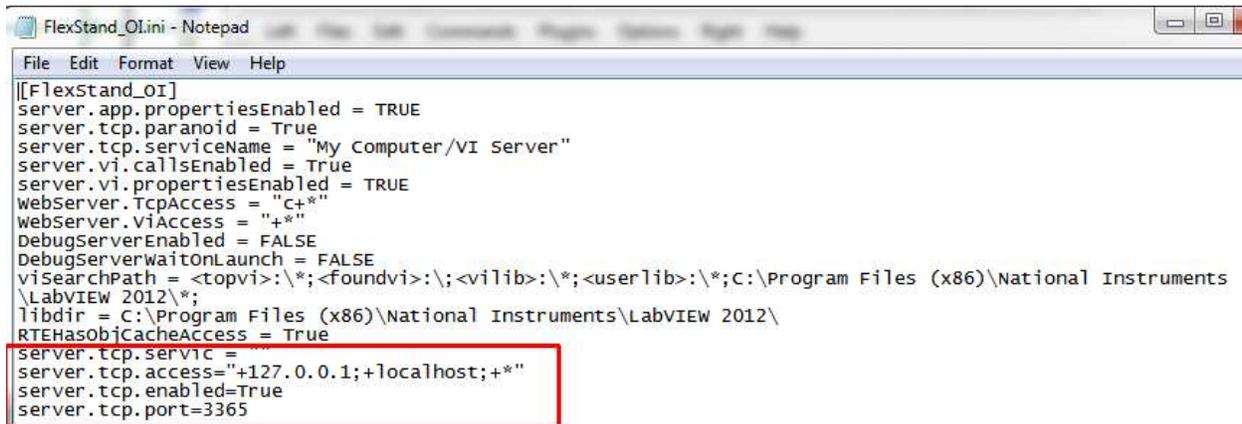
The contents of the VI is:



This VI calls the buffer, which is located in another execution instance (TestStand Execution engine). In the other instance the VI needs to be in memory.
Idea from: <http://screencast.com/t/zRDGaScM>

Network settings

VI server must be enabled in the FlexStand_OI.ini file:



```
FlexStand_OI.ini - Notepad
File Edit Format View Help
[[FlexStand_OI]
server.app.propertiesEnabled = TRUE
server.tcp.paranoid = True
server.tcp.servicename = "My Computer/VI Server"
server.vi.callsEnabled = True
server.vi.propertiesEnabled = TRUE
webServer.TcpAccess = "c+*"
webServer.ViAccess = "+*"
DebugServerEnabled = FALSE
DebugServerwaitOnLaunch = FALSE
viSearchPath = <topvi>:\*; <foundvi>:\; <vilib>:\*; <userlib>:\*; c:\Program Files (x86)\National Instruments
\LabVIEW 2012\*;
libdir = C:\Program Files (x86)\National Instruments\LabVIEW 2012\
RTEHasObjCacheAccess = True
server.tcp.servicename = "My Computer/VI Server"
server.tcp.access="+127.0.0.1;+localhost;+*"
server.tcp.enabled=True
server.tcp.port=3365
```

It is also important that no firewalls blocks the communication.