

10 Tips & Tricks on how to use Rohde & Schwarz® LabVIEW™ Instrument drivers Application Note

Having the possibility to remotely control instruments has become the necessity not only in the field of automated production testing but more and more already in the phase of development. One of the programming environments that makes this task accessible for designers with very little or no programming experience is LabVIEW. To make the remote control of instruments easier, Rohde & Schwarz provides LabVIEW Instrument drivers that take away the burden of synchronization, response formatting and error handling from the developer.

This Application Note gives 10 useful Tips & Tricks for the users of Rohde & Schwarz instrument drivers. For demonstration, we chose the Spectrum Analyzer rspecan instrument driver version 3.8.0.

Microsoft® and Windows® are U.S. registered trademarks of the Microsoft Corporation.

R&S® is a registered trademark of Rohde & Schwarz® GmbH & Co. KG.

National Instruments® are U.S. registered trademarks of National Instruments.

LabVIEW™ is a trademark of National Instruments.

Note:

Please find the most up-to-date Application Note on our homepage:

www.rohde-schwarz.com/appnote/1MA228

Instrument Drivers LabVIEW – 1MA228_5e

Contents

1	Mass Compile Drivers before Use.....	3
2	Speed-up Your Development with SCPI Searcher Plug-in.....	4
3	Leave the Boring Wiring to Connect&Align Plug-in.....	9
4	Use the Pattern Initialize -> Close -> Error Message.....	12
5	Always Use the Single Mode.....	13
6	Take Advantage of Utility VIs.....	14
7	Keep the VISA Timeout Low.....	16
8	Get as Fast as Possible.....	17
9	Use ExpressVI Configurator.....	19
10	Building Executables.....	23
11	Rohde & Schwarz.....	25

1 Mass Compile Drivers before Use

All Rohde & Schwarz instrument drivers are compiled in LabVIEW2010 and can be used in LabVIEW 2010 or newer. Mass compilation assures that the entire driver is compiled for the LabVIEW version you are using. New self-extracting installers perform mass compilation automatically, but you can always start it manually:

Menu-> Tools -> Advanced -> Mass Compile, choose a root directory to compile, e.g.:

`instr.lib\Rohde&Schwarz Spectrum Analyzer\`

and press **Mass Compile** button.

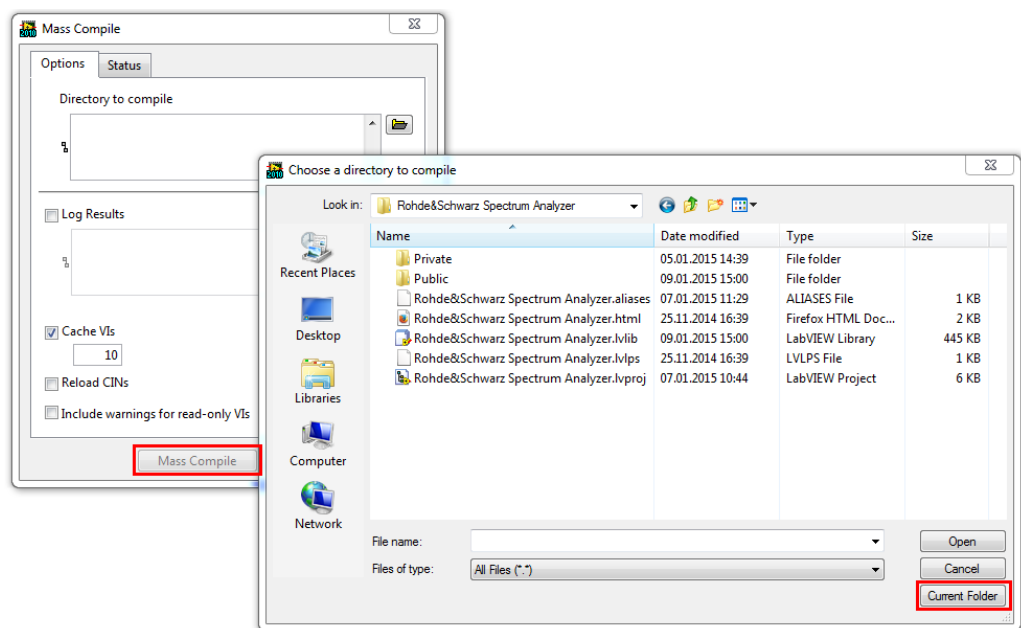


Figure 1-1: LabVIEW Mass Compilation of the driver

2 Speed-up Your Development with SCPI Searcher Plug-in

SCPI Searcher plug-in provides many useful features when developing your remote-control application. Here are the key ones:

- Finding a driver VI or a driver attribute that is using SCPI command(s) of your interest.
- Importing an IO trace (NI IO-Trace or R&S VISA IO Log) SCPI commands and searching the driver functions. If an IO trace contains communication with more than one device, a filter can be used to assign the commands to the proper driver or leave them out.
- Importing an C/C++ code that uses R&S VXIplug&play instrument driver and searching for equivalents in LabVIEW drivers - this feature allows for a quick conversion of an existing C/C++ measurement task to LabVIEW.
- Importing R&S CMW Run Result xml data and search for the same functions in LabVIEW drivers.
- Performing a test of an entire list with SCPI commands for their presence in the LabVIEW driver.



Since the LabVIEW driver functions or attributes have the same names in CVI / VXI-plug&play drivers, you can practically use this plugin to search for a CVI/VXIplug function as well.

Installation: Download and run the `rsidrToolboxInstall.vi` that is available as an attachment to this application note:

<http://www.rohde-schwarz.com/appnote/1MA228>

Launching the plug-in: In your LabVIEW Block Diagram, start the LabVIEW **Quick Drop** by pressing **CTRL+SPACE** and then **CTRL+F** to launch the SCPI Searcher plug-in.



The **first** launch of the LabVIEW **Quick Drop** can take several seconds, due to the population of all available LabVIEW palettes and functions.

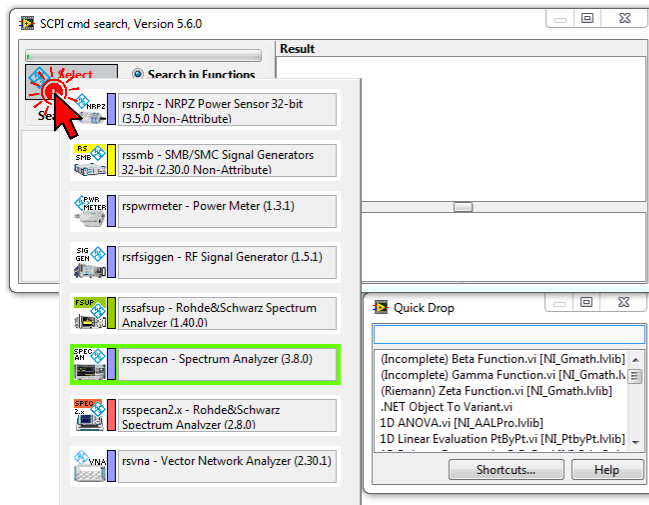


Figure 2-1: SCPI Searcher window.

Select a driver by clicking the driver icon. The plugin displays all the installed drivers. If you start to type, it works as filter on driver names. In our list of drivers typing **spec** leaves only the Spectrum Analyzer items:

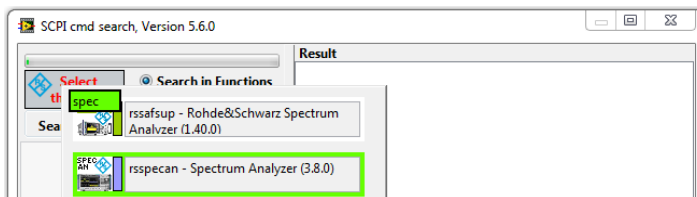


Figure 2-2: Filtered driver selection window.

After selecting the driver, paste a SCPI sequence to the left **Search text field**. For demonstration, we use a sequence from the FSW user manual (Chapter "Programming examples"):

```
//-----Preparing the measurement -----
*RST
//Resets the instrument
INIT:CONT OFF
//Selects single sweep mode.
//-----Configuring the Frequency and Span-----
FREQ:CENT 100MHz
//Defines the center frequency
FREQ:SPAN 200MHz
//Sets the span to 100 MHz on either side of the center frequency.
//Activate signal tracking to keep the center frequency on the signal peak:
CALC:MARK:FUNC:STR ON
CALC:MARK:FUNC:STR:BAND 20MHz
CALC:MARK:FUNC:STR:THR -90dBm
CALC:MARK:FUNC:STR:TRAC 1
```

Make sure that you have **Search for the SCPI** selected - this uses tailor-made searching algorithm that fits SCPI commands canonical and short forms. After pasting the text, right-click on that **Search text field** and select **Test -> Test All**:

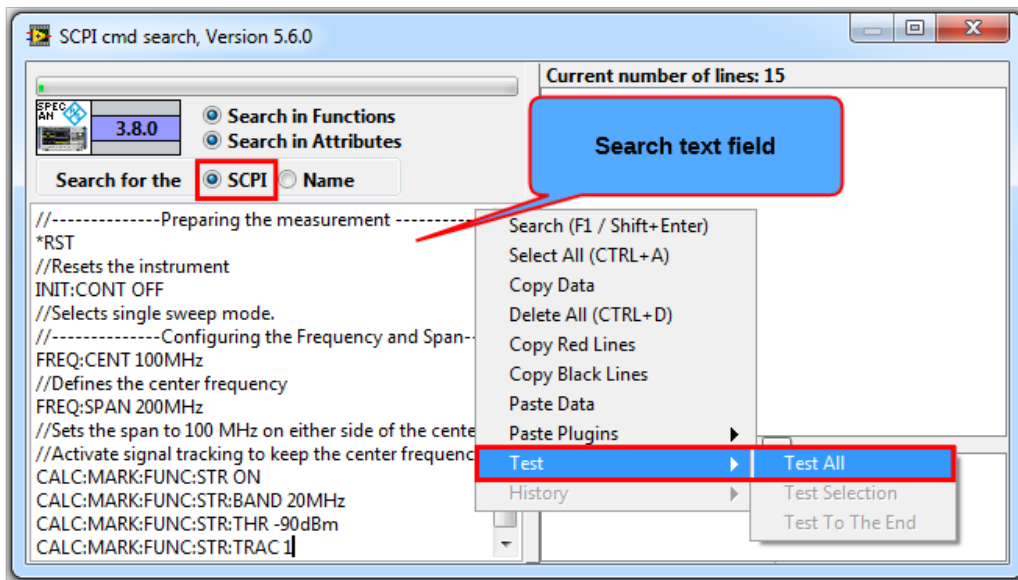


Figure 2-3: Pasting a SCPI command sequence and testing and for known SCPI commands.

The red marked lines were the ones not found:

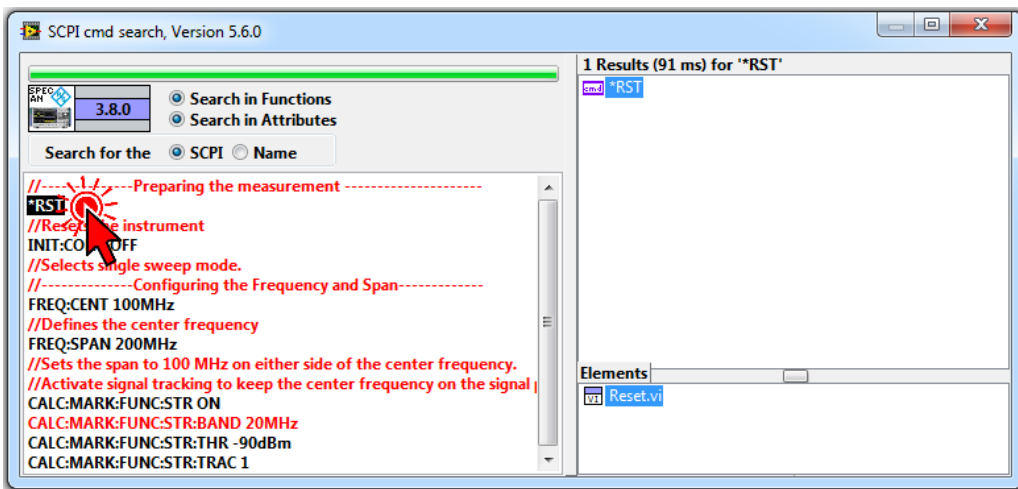


Figure 2-4: Result of batch testing of all the lines in the search field.

The red lines are mostly comments, but there is also one non-found SCPI command among them. Now you can double-click on any individual line to search for driver items.



Search for an entire command by placing the cursor at the end of the line and: **double-click**, hit **F1** button or **SHIFT+ENTER** buttons combination. If you do not want to search for the entire command, place the cursor accordingly. The search is then made from the beginning of the line up to the cursor.

To search for any part of a string, make a text selection and use the **F1** or **SHIFT+ENTER**.

In our example, searching for the ***RST** matched the ***RST** command and in **Elements** you see its usage in the driver function `Utility\Reset.vi` Now you can simply drag and drop the `Reset.vi` element to your Block Diagram. Dragging an attribute name to your Block Diagram opens ExpressVI Configuration window (see **Tip 9**). After you have done the similar procedure for all the commands of interest, you can use the **Connect&Align** plugin explained in the **Tip 3**.

2.1 Importing IO Traces

If you want to extract a SCPI sequence from an IO trace, right-click on the **Search text field** and select **Paste Plugin -> Import IO Trace**. Supported formats are NI IO Trace (exported as txt file) and R&S VISA Log file. As an example, we use the file `IOtraceFSW+SMW.txt` that you find in the attachment to this application note:

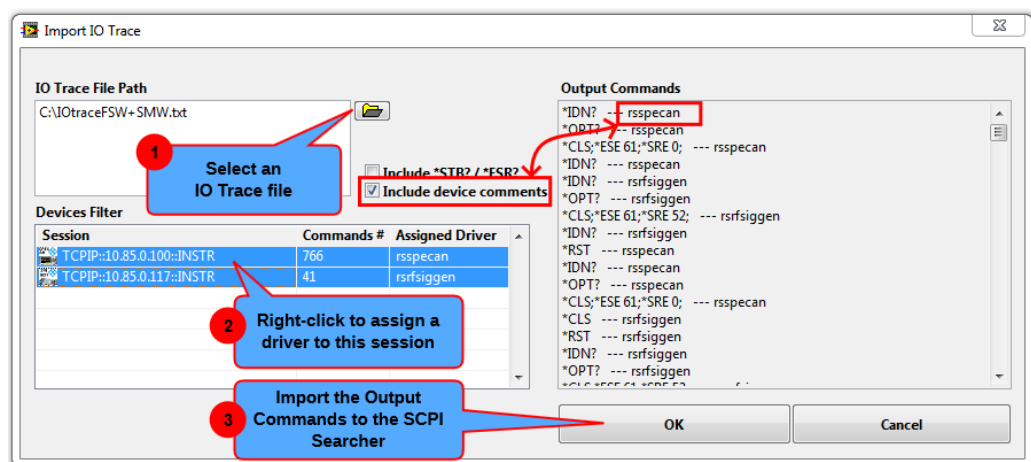


Figure 2-5: IO Trace Import Window

Select the IO Trace file with the Browse button. After parsing of the file is finished, you have a choice of assigning Sessions to drivers. In our case, the IP address `10.85.0.100` is an FSW and `10.85.0.117` is an SMW. By default all the sessions are selected to be exported into the **Output Commands** list, but you can change it in the **Devices Filter** table by selecting / deselecting the rows. Checking the box **Include device comments** adds a tag after a command (e.g. `--- rsspecan`) to let the SCPI Searcher know to which driver the command belongs. The advantage is, that the order of commands is exactly as they were in the imported IO Trace. Hitting OK imports the **Output Commands** to the SCPI Searcher for further operations.

2.2 Other Importing Filters

Import Plugins -> Paste CVI Functions - copy an existing C/C++ code with R&S VXI-plug&play drivers to your Clipboard and use this import filter to create LabVIEW code performing the same actions as your existing C/C++ code.

Import Plugins -> Import CMW Run XML Log - extract SCPI commands used by the CMW Run and create your LabVIEW application equivalent.

Import Plugins -> Paste CALL InstrWrite - some of R&S user manuals (e.g. FSV / FSVA) show remote-control application examples using the following syntax:

```
CALL InstrWrite (analyzer, "*CLS")
```

Copy such user manual example to your Clipboard, and use this menu item to parse out SCPI commands.

Import Plugins -> Paste LOC Lines - copy R&S user manual List of commands to your Clipboard and use this menu to parse SCPI commands. Example of LOC lines from the FSW user manual:

List of Commands	
[SENSe:]WINDow:DETEctor<trace>[:FUNCTION].....	860
[SENSe:]WINDow:DETEctor<trace>[:FUNCTION]:AUTO.....	861
[SENSe:]ADJust:ALL.....	777
[SENSe:]ADJust:CONFIguration:HYSTeresis:LOWer.....	778
[SENSe:]ADJust:CONFIguration:HYSTeresis:UPPer.....	779
[SENSe:]ADJust:CONFIgure:LEVel:DURation.....	778
[SENSe:]ADJust:CONFIgure:LEVel:DURation:MODE.....	778

Only these parts will be extracted

Figure 2-6: User Manual List Of Commands

3 Leave the Boring Wiring to Connect&Align Plug-in

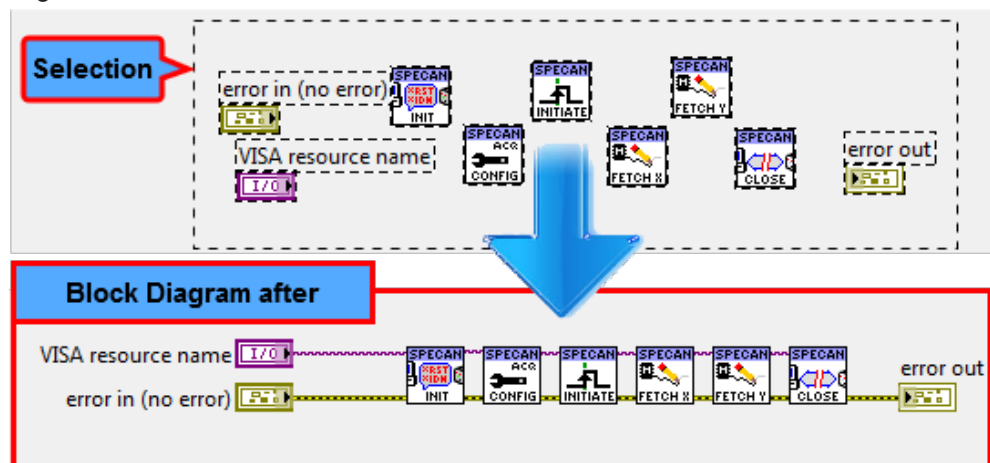
This plug-in speeds up tedious tasks of connecting your driver VIs together with VISA and error cluster wires. It takes X-coordinates of Block Diagram objects in your selection, and aligns them into a chain by moving them and making horizontal space. Here are the main features:

- Order of connection is only defined by the X-coordinates objects in the selection.
- It works not only on the driver VIs, but on all nodes with error in/out terminals. VISA terminals are optional.
- If a node has no VISA terminal, the plugin connects error terminals and keeps searching for the next node with matching VISA terminal to connect (see the example pictures below).
- The plug-in only connects VISA terminals of matching drivers, hence avoids mistakes of connecting wrong session and driver together.
- By default, the Y-alignment is done according to the most-left object in the selection. Using **CTRL+SHIFT+A** changes that to the most-right selected object.
- The plugin also works without any nodes - just on tunnels and/or control/indicators.
- Wiring through tunnels is done in a smart way - if a tunnel is in your selection, it is reused instead of creating a new one. This feature avoids creating parallel tunnels that are hard to recognize.
- If a Case Structure Selector is in the selection, error wire is rewired through it, and all existing connections inside all case diagrams are maintained. The same rule applies for Shift Registers.

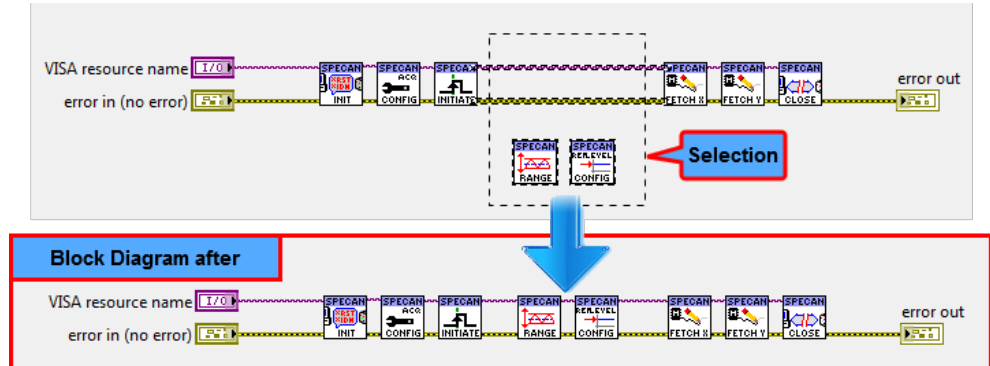
Installation: See the **Tip 2: SCPI Searcher Plug-in**

Below, are some before and after pictures showing the plug-in capabilities. First, you need to perform a selection either by area or selecting individual objects. Then invoke the LabVIEW **Quick Drop** by **CTRL+SPACE**, and run the plug-in by **CTRL+A**:

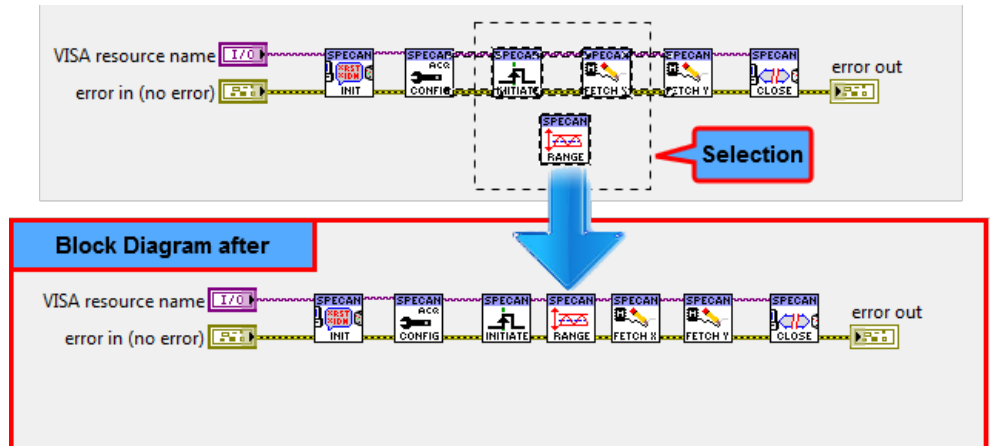
- Connecting driver VIs together. The left-most error terminal object is used for Y-alignment:



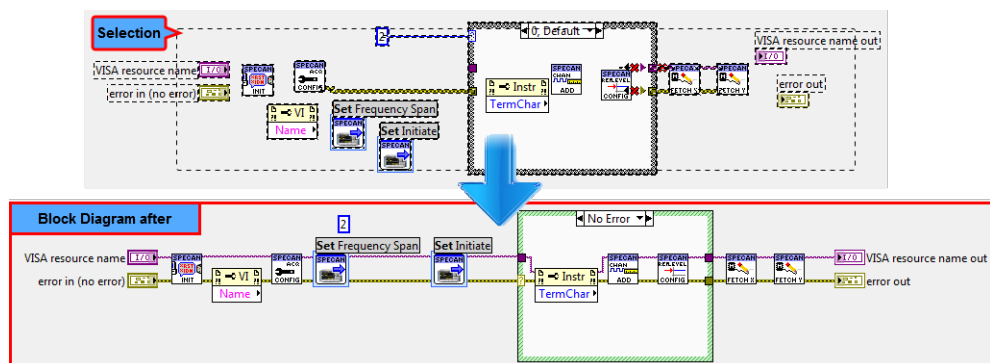
- Selecting new nodes and error / VISA wires that start left to the selection and end right to the selection inserts the VIs into the existing chain:



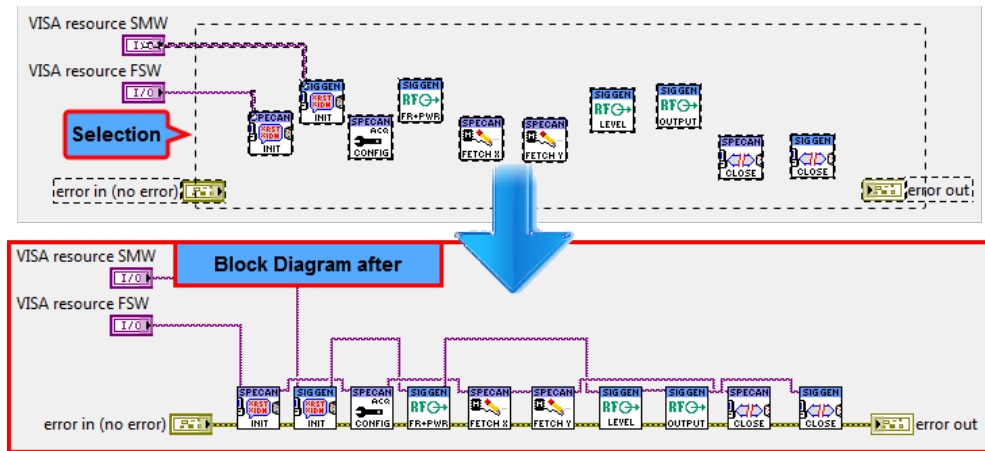
- Adding a VI in the middle of an existing chain by selecting also left and right node. All the nodes on the right are shifted to make required space:



- Connecting through a Case Structure with its case selector terminal included in the selection. The error wire is wired through the case selector terminal. Notice the VISA tunnels being reused and thus maintaining the other cases of the Case Structure properly connected. The Property node without VISA input terminal is still connected to the error wire:



- Connecting different drivers together keeps VISA sessions properly assigned:



4 Use the Pattern Initialize -> Close -> Error Message

Although R&S drivers use standard VISA session, make sure that you open a new session with `Initialize.vi` instead of `VISA Open` function of LabVIEW. This makes sure all the necessary additional data required by the instrument driver are initialized as well.

To close the remote-control session, call the `Close.vi` to clean up the driver session data. If you call the `Initialize.vi` twice with the same resource name, it opens a new connection to the instrument, but safely reuses previously initialized session data.

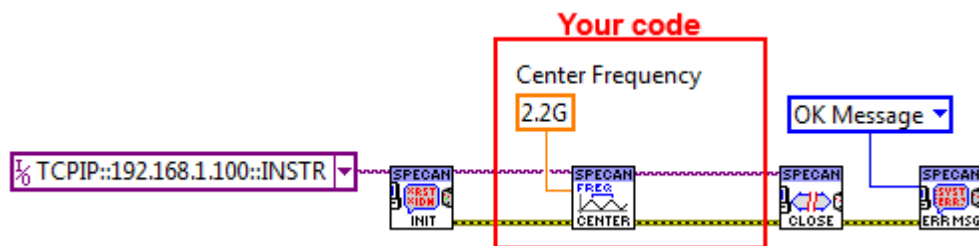


Figure 4-1: Standard instrument driver application pattern

Function `Error Message.vi` can be configured to show a Dialog window if an error occurs, or just to provide a decoded message string output. VISA session input is not mandatory, it is internally wired through.



Error cluster string parameter contains additional information about error that occurred often with suggestions on how to fix it. The entire error call chain is also part of the string.

5 Always Use the Single Mode

The **single** most common source of problems when remote-controlling measurement instruments is wrong measurement synchronization mostly caused by an instrument not being in **single** measurement mode. The symptoms are unpredictable or non-repeatable results that a programmer is trying to solve by inserting fixed time delays.

By default, all Rohde & Schwarz Spectrum Analyzers operate in continuous measurement mode. You have to switch them to **single** mode after session initialization, reset or after changing measurement application.

The topic of **single** vs continuous mode is relevant for instruments that perform an acquisition task: Spectrum Analyzers, Network Analyzers, Oscilloscopes, Communication Testers, Power Meters, Audio Analyzers, and so on... Let us call them - acquisition class instruments. For these instruments we can extend the default pattern mentioned in **Tip 4** to:

Initialize -> Single Sweep -> Close -> Error Message

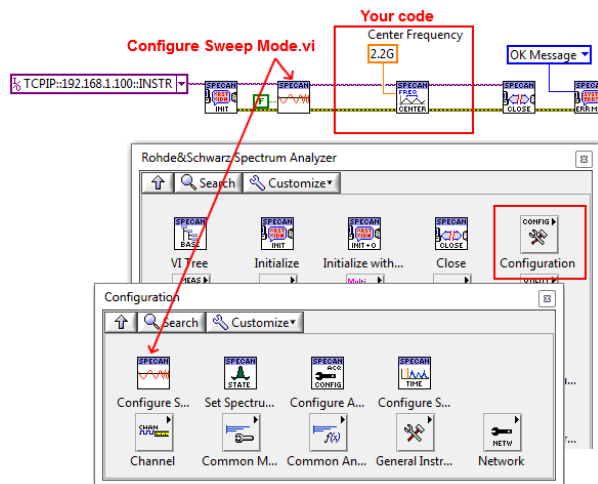


Figure 5-1: Spectrum Analyzer Configure Sweep Mode VI



Switching to different instrument option application on the FSW e.g. FSW-K40 Phase Noise Measurement switches back to continuous measurement mode. Always make sure you switch to the desired instrument application first, and only then set the acquisition mode to **single**.



Driver VIs with '**Read**' in their names start new acquisition, wait for it to finish and then return results. If you do not want to start a new acquisition, use driver's '**Fetch**' VIs. They only return previously acquired values.

6 Take Advantage of Utility VIs

Here are some of the interesting VIs in the Utility group:

6.1 Instrument IO VIs

The Palette offers VIs for sending direct SCPI commands or queries. Use these VIs when you cannot find a driver function or an attribute with the SCPI Searcher.

Another common operation is transferring files from the instrument to your PC, e.g. screenshot files. For this purpose use the:

`Read To File From Instrument.vi`

To transfer files in the opposite direction (PC -> Instrument) e.g. Setup files, use the:

`Write From File To Instrument.vi`

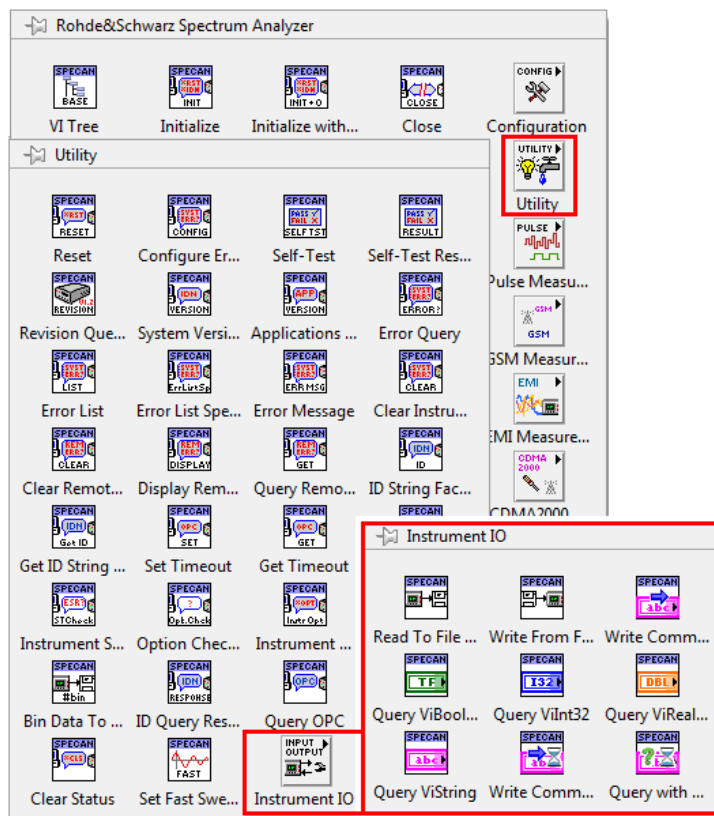


Figure 6-1: Instrument IO VIs on the Utilities Palette

6.2 Configure Error Checking.vi

The VI configures types of check mechanisms that the instrument driver performs. By default, after initializing the session all three error checking features are switched **ON**.

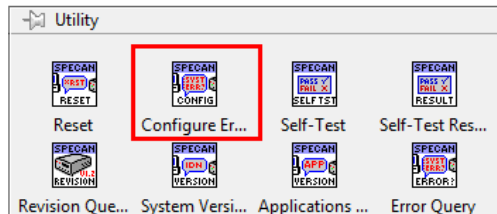


Figure 6-2: Configure Error Checking.vi

Switch the **Option Checking** OFF when an instrument driver generates the error '**Option is not installed**', although you are sure that the functionality is available.

Switch the **Range Checking** OFF when an instrument driver generates the error '**Value out of Range**', although you are sure that the value is accepted by the instrument.



Status Checking feature: By default, instrument drivers query instrument's Status Byte Register (***STB?** query) and if necessary also send **'SYST:ERR?'** query. This feature is convenient since you are immediately informed about the status of your instrument. This action takes small, but sometimes critical amount of time (see **Tip 8**). Therefore it can be switched OFF by setting the **Status Checking** to **OFF**.

We recommend keeping the **Status Checking** **ON** as much as possible, since it provides constant assurance that your remote-control application is not producing any instrument errors.

6.3 Query OPC.vi

If some instrument settings take a longer time, sometimes it is necessary to make sure that the instrument has finished all pending operations. Using this VI causes your program to wait until the instrument reports its readiness to continue.

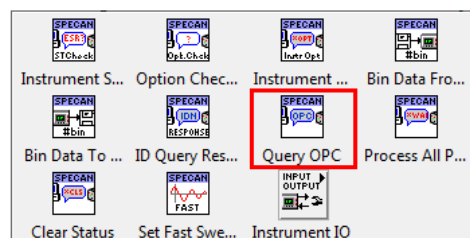


Figure 6-3: Query OPC.vi on the Utilities Palette

7 Keep the VISA Timeout Low

Waiting for an instrument to respond over VISA in LabVIEW **cannot be interrupted**. It has to either end up with instrument responding, or resulting in **VISA Timeout**. During this time, your application becomes non-responsive. Therefore keeping the **VISA Timeout** as low as possible to avoid long non-interruptible periods in your program.

Rohde & Schwarz instrument drivers are using so-called **OPC Timeout** for operations where longer response times are to be expected. A typical example of such case is waiting for Spectrum Analyzer to finish a sweep acquisition or performing instrument self-calibration. The **OPC Timeout** is settable, and only used by drivers internally.



Due to the reason mentioned above trying to increase the **VISA Timeout** to prevent long-lasting operations resulting in timeout error will not help you solve the problem. You have to change the **OPC Timeout** instead.

Set / Get OPC timeout VIs are available on the Utilities Palette. For setting or getting the **VISA Timeout** you have to use the LabVIEW VISA palette:

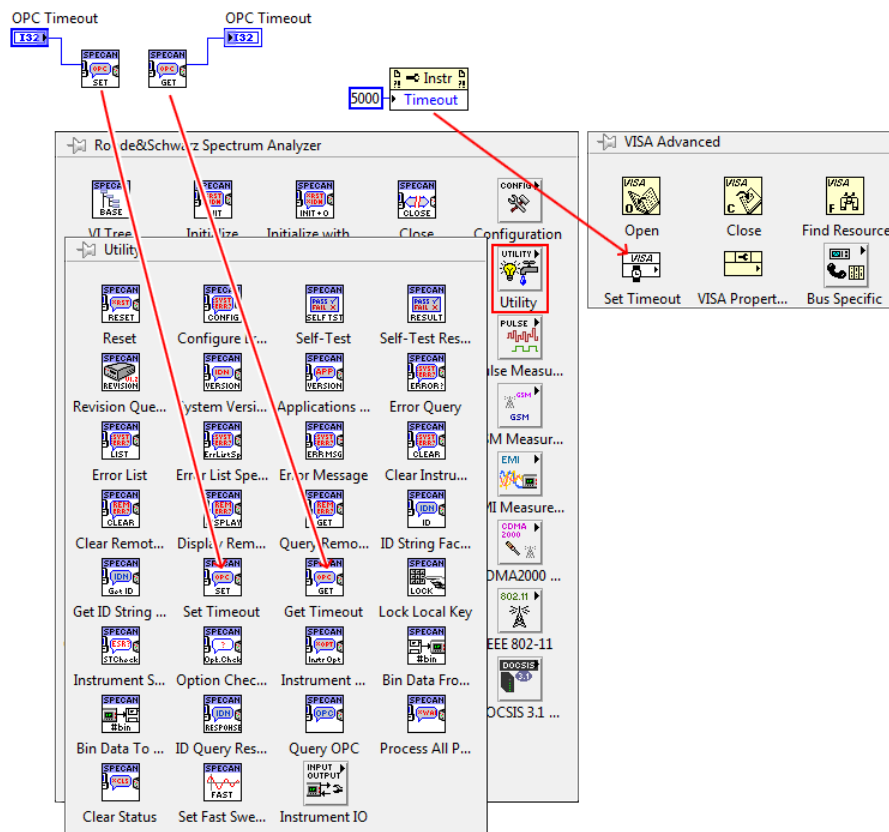


Figure 7-1: Instrument driver OPC Timeout VIs vs VISA Timeout property node

8 Get as Fast as Possible

One of the most common arguments against using LabVIEW is lack of performance. Is it true when remote-controlling instruments? The detailed elaboration on this topic can alone fill an entire application note. To evaluate this, we choose a simple task that is inevitably found in every remote-control application using Spectrum Analyzers - performing a sweep acquisition. This is repeated several times to get an averaged result. The actual **Tip 8** comes at the end of this chapter.

We perform the same sweep acquisition task in four different ways:

- LabVIEW 2010 x86 with just using VISA Read / VISA Write function, no instrument driver involved - Orange bar
- Visual Studio 2015 C++ program as a reference to compare LabVIEW to other development environments - Blue bar
- Rohde & Schwarz Spectrum Analyzer instrument driver ver. 3.8.0 with Error Checking switched OFF - Green bar
- Rohde & Schwarz Spectrum Analyzer instrument driver ver. 3.8.0 with Error Checking switched ON - Violet bar

VISA Component: visa32.dll, Version 16.0, National Instruments

Instrument: FSW26 FW 2.50 SP1, set to Zero Span sweep of 10ms sweep time. The graph below shows on the x axis the duration of one acquisition cycle (averaged over 1000 repetitions):

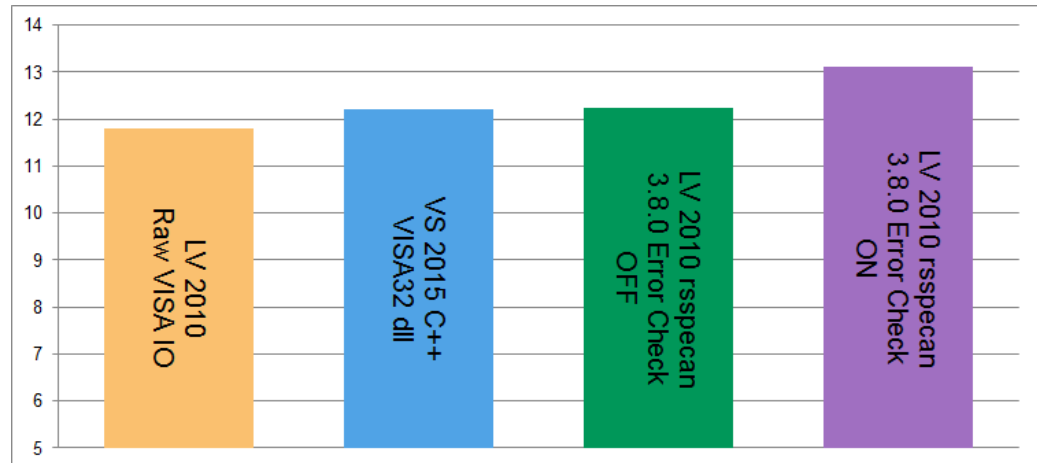





Figure 8-1: Duration [ms] of one acquisition cycle of 10ms sweep time

It is obvious that LabVIEW is as fast as C++ program (here, slightly faster). Although it might not be as performant in operating Graphical User Interfaces or dealing with complex cluster-data structures, for instrument remote-control it matches performance of other alternative programming environments. **The instrument drivers add only c.a. 4 per cent of overhead; see ■ versus ■**



The difference between the  and  shows the overhead of error checking performed by the instrument drivers. As mentioned in **Tip 6**, you can switch this feature OFF for the sections of your program requiring maximum speed. Then, outside those sections, switch it back ON to make sure that you are communicating with your instrument error-free.



Keep in mind that the error checking overhead is constant. Therefore, if in our tests we changed the FSW sweep time to, e.g. 100ms, the  would practically be of the same height as the other three.

In other words: ***For longer acquisition times, the error checking overhead is negligible and can be left ON all the time.***

9 Use ExpressVI Configurator

ExpressVI Configurator allows you to access any attribute of a driver. An attribute is a single parameter of an instrument that can be set or read from an instrument. A typical example of an attribute is Spectrum Analyzer Center Frequency (called `RSSPECAN_ATTR_FREQUENCY_CENTER`). The ExpressVI Configurator is used by instrument driver developers to combine more attributes into so-called **High-level** functions. In our example, the `RSSPECAN_ATTR_FREQUENCY_CENTER` is used in the following High-level functions (among others):

- `Configure Frequency Center.vi`
- `Configure Frequency Center Span.vi`
- `Configure Phase Signal Settings.vi`

Why should you use ExpressVI Configurator? There are several reasons to do so:

- **Reading parameter value:** There is no High-level function to read the Center Frequency back from the instrument. You have to use the ExpressVI configured to Read operation.
- **Setting just one parameter separately:** A High-level function may contain setting of several attributes. Sometimes to achieve an instrument setting you require, it is necessary to use a different order of commanding the attributes or setting a particular attribute only.
- **Non-availability of a High-level function:** To minimize driver's size, High-level functions that set just one attribute are often missing. If you cannot find a High-level function using the attribute you need, you always have the option to configure the ExpressVI and use it in your code.

To access the ExpressVI configuration window, do one of the following:

- Double-click an existing already configured ExpressVI
- Copy/Paste an existing ExpressVI and double-click on it
- Drag an attribute name found by the SCPI Searcher and drop it to your Block Diagram
- Select it from a driver's Palette:

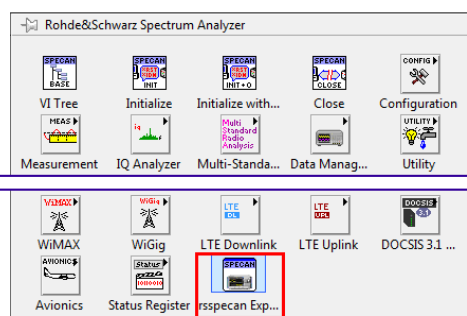




Figure 9-1: ExpressVI on the driver's Palette

The ExpressVI configuration window shows the attributes in a tree structure exactly like in the attributes `chm` help file:

Search for an attribute with different criteria or select it from the attribute tree control. Then use the Attribute Read  or Write  operation:

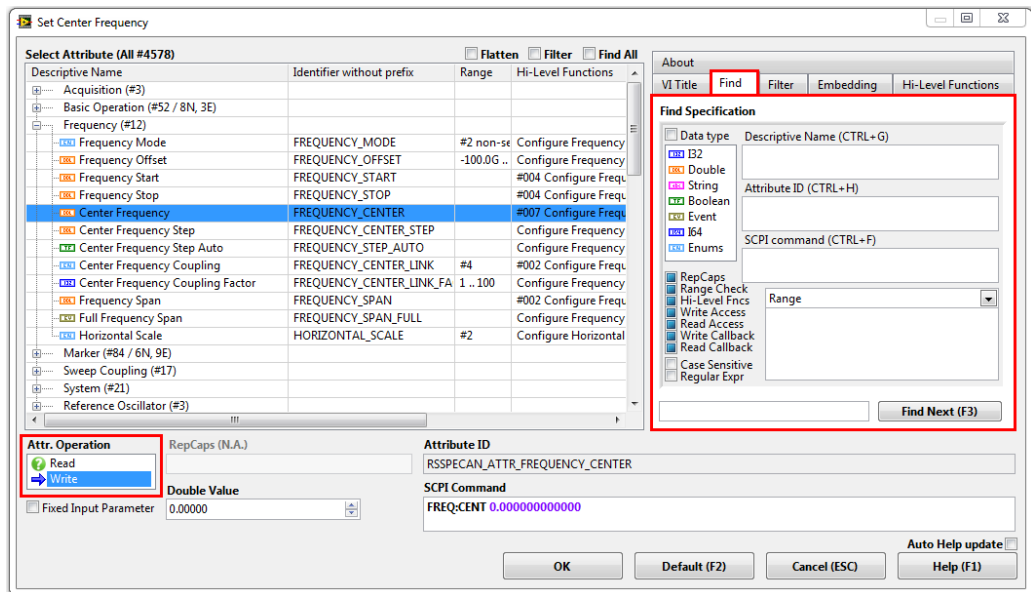


Figure 9-2: ExpressVI configuration window

You can also see in which High-level function the attribute is used. With the right-click context menu you can place the High-level function VI on your Block Diagram or to the Clipboard:

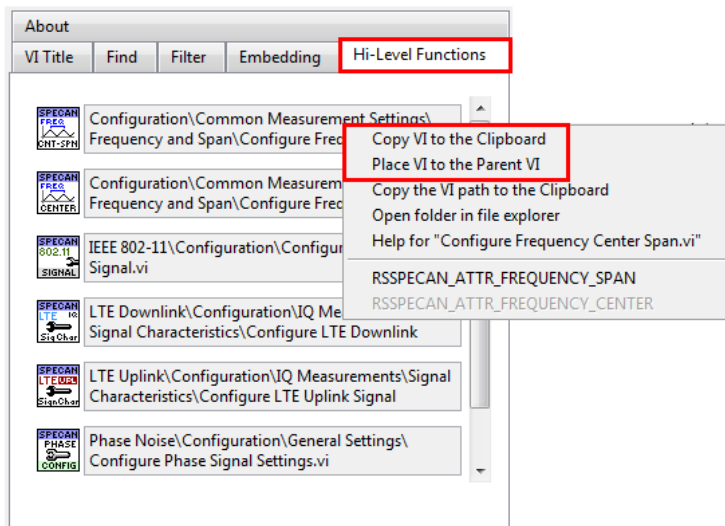


Figure 9-3: High-level functions using the selected attribute

Let us take the attribute `RSSPECAN_ATTR_MARKER_ENABLED` to briefly explain **Repeated Capabilities (RepCaps)**: This attribute has Boolean value of True or False (Marker State ON or OFF). However, you also need to communicate to the instrument which one of the sixteen markers you want to enable. This information you enter as **Repeated Capability** string:

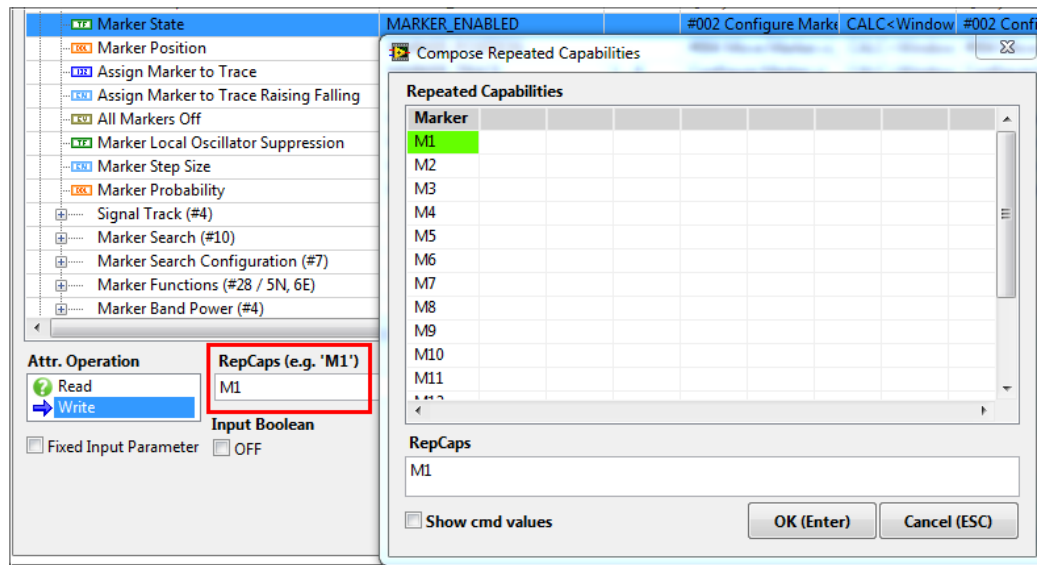


Figure 9-4: Composing Repeated Capabilities in the ExpressVI Configurator

Double-click on the empty RepCaps string control (marked with red frame above) and it will be filled with the first available value. Double-clicking on it again opens **Compose Repeated Capability** window as shown on the picture above. If an attribute have more Repeated Capabilities defined, they are separated by **comma** without spaces.

If you do not intend to make the Repeated Capability value variable for your program, you can enter its constant value here. For example if you only want to address **Marker 1**, enter **M1** in the **RepCaps** and it becomes default value. If you leave the field **RepCaps** empty, you are required to connect the ExpressVI Repeated Capabilities terminal in your Block Diagram.

For convenience and to avoid errors when composing Repeated Capabilities strings, the ExpressVI Configurator offers the **Variable RepCaps** feature. Let us assume you want to make a marker number variable. You must compose the Repeated Capabilities string:

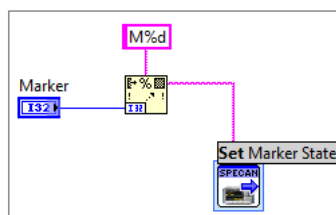


Figure 9-5: Composing Repeated Capabilities as string

Here is the same functionality achieved by using Variable RepCaps feature of the ExpressVI Configurator. In this case, the **RepCaps** string control must be filled with a correct value, because it is used as a default one:

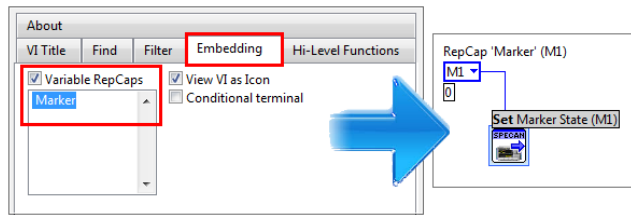


Figure 9-6: Variable RepCaps available in Embedding Tab

If you want to perform an attribute operation conditionally, you can use the True/False Case Structure or select the **Conditional terminal** checkbox of the ExpressVI Configurator:

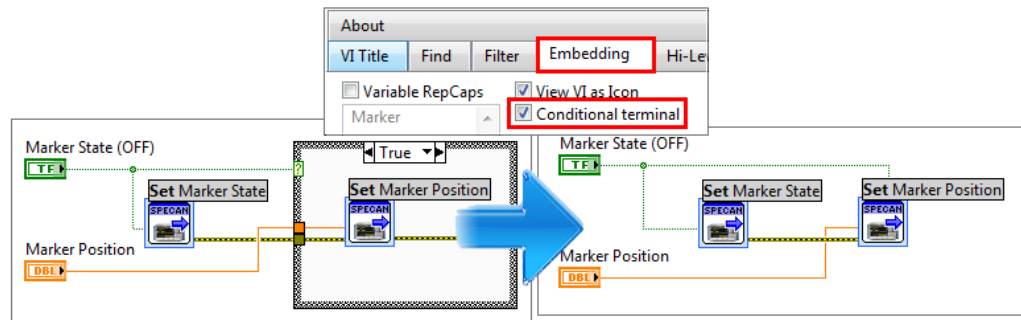


Figure 9-7: Using Conditional terminal for ExpressVI instead of wrapping it in True / False Case Structure

If you want to change the default ExpressVI Label, go to the VI Title Tab and make the changes. Right-click on the **VI Title compose string** and from the context menu select **Make all as default**:

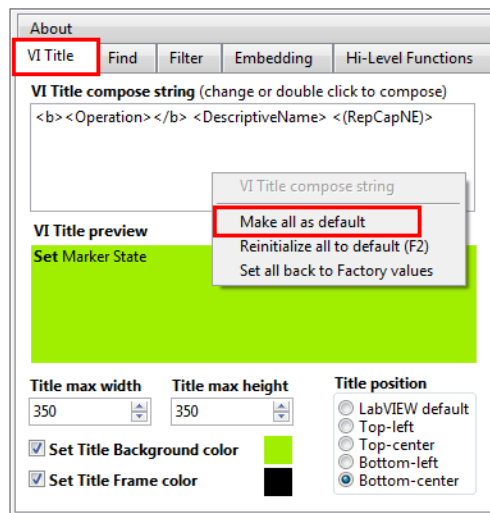


Figure 9-8: Configuring the default ExpressVI Title

All the VI title settings are stored and used for every new ExpressVI further on. Previously configured ExpressVIs are not affected. You must open their configuration panel again and use the **Default (F2)** button.

10 Building Executables

Rohde & Schwarz instrument drivers use dynamic calls for VIs placed in the driver `instr.lib` folder:

```
_utility\callbacks
```

or

```
Private\callbacks
```

When running in development environment, those VIs are always available. When you create an executable however, none of these callbacks are in your Startup VI dependency list, therefore they are not included into the final executable application. As a consequence, if you run your exe program, the first time the driver needs to make a dynamic VI call, it reports an error `-1074003958 (0xBFFC000A)` with the following text: `Cannot find dynamically called callback '<callback_name>'`. Make sure that all VIs are in correct folders. If deploying, folder structure has to be maintained!

To solve this error, make sure that if you are using an instrument driver that is organized in a library, e.g. the Spectrum Analyzer driver, you have the corresponding `lvlib` file added into your project:

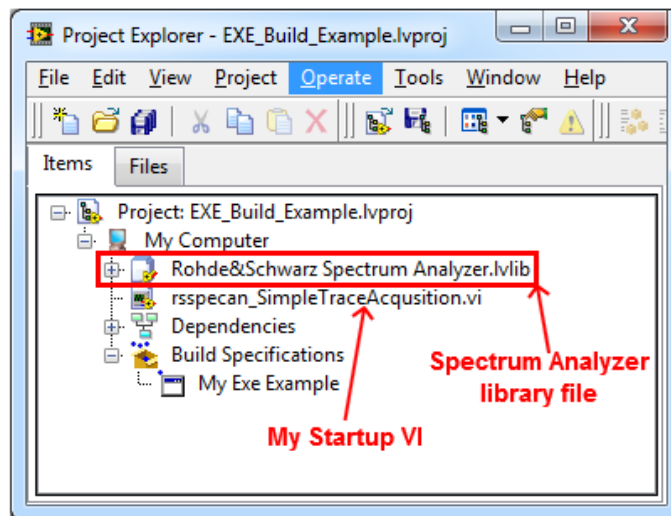


Figure 10-1: Example of a project that builds the executables

Open your Project Build Specification and add the callbacks folder to the **Always Included** list:

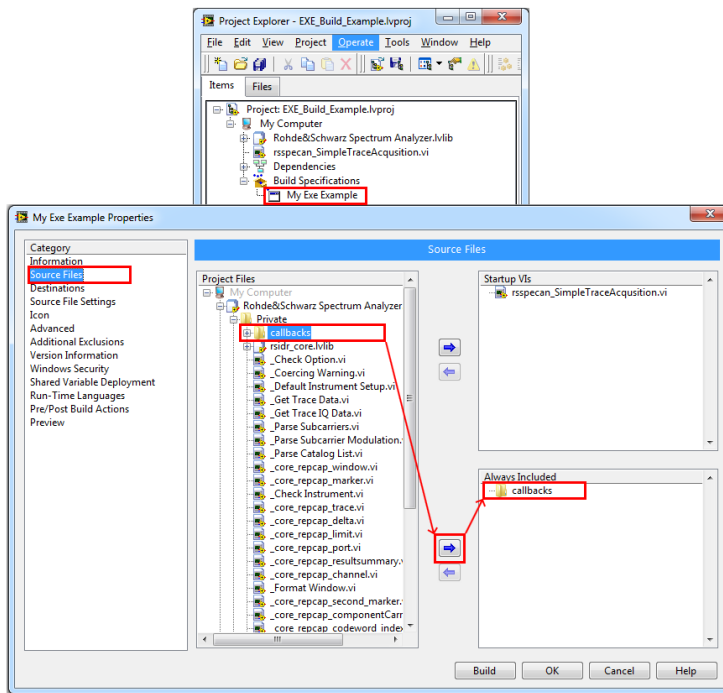


Figure 10-2: Building an executable requires adding entire callbacks folder to Always Included list

If you are using more than one driver, you need to repeat this procedure for each of them.

Another common problem when building executables are long filenames. Although under Windows 10 the limitation of 255 characters for a file path has been removed, with older Windows versions you have to check the box **Use LabVIEW 8.x file layout**:

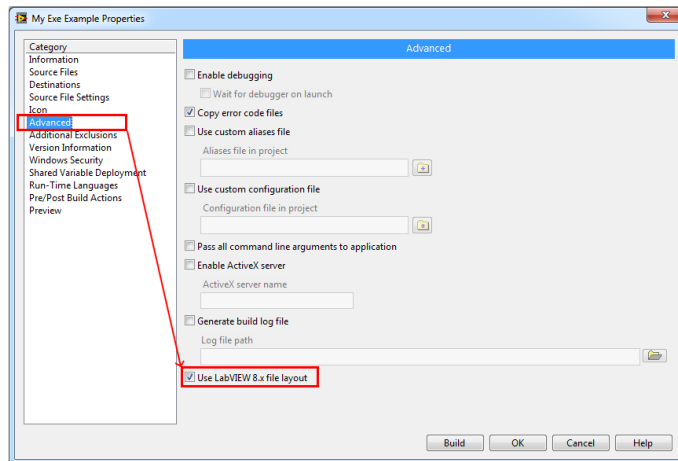


Figure 10-3: Solution for problems with long file paths

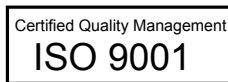
The EXE_Build_Example.lvproj file is available in the attachment to this application note.

11 Rohde & Schwarz

Rohde & Schwarz is an independent group of companies specializing in electronics. It is a leading supplier of solutions in the fields of test and measurement, broadcasting, radiomonitoring and radiolocation, as well as secure communications. Established more than 80 years ago, Rohde & Schwarz has a global presence and a dedicated service network in over 70 countries. Company headquarters are in Munich, Germany.

Sustainable product design

- Environmental compatibility and eco-footprint
- Energy efficiency and low emissions
- Longevity and optimized total cost of ownership



Regional contact

- Europe, Africa, Middle East
Phone +49 89 4129 12345
customersupport@rohde-schwarz.com
- North America
Phone 1-888-TEST-RSA (1-888-837-8772)
customer.support@rsa.rohde-schwarz.com
- Latin America
Phone +1-410-910-7988
customersupport.la@rohde-schwarz.com
- Asia/Pacific
Phone +65 65 13 04 88
customersupport.asia@rohde-schwarz.com
- China
Phone +86-800-810-8228 / +86-400-650-5896
customersupport.china@rohde-schwarz.com

Headquarters

Rohde & Schwarz GmbH & Co. KG

Mühldorfstraße 15 | D - 81671 München

+ 49 89 4129 - 0 | Fax + 49 89 4129 - 13777

www.rohde-schwarz.com

This application note and the supplied programs may only be used subject to the conditions of use set forth in the download area of the Rohde & Schwarz website.

R&S® is a registered trademark of Rohde & Schwarz GmbH & Co. KG. Trade names are trademarks of the owners.