OptiInstrument

OptiInstrument 2.0 Release Notes

IMPORTANT

- Before installing OptiInstrument, make sure that NI-VISA.NET Runtime is installed on the system. When installing the NI-VISA, select the NI-VISA.NET Runtime.
- Users can install NI-VISA from the following link https://www.ni.com/en-ca/support/downloads/drivers/download.ni-visa.html#346210.

  Note: The NI-VISA is a large package, it may take longer time to download/install.

Installation Notes:

- When starting the installation process of OptiInstrument, the popup message shown in Fig.1 offers the user a choice to quit the installation process if the NI-VISA.NET Runtime package is not installed on the same computer or continue the installation if the package is installed. However, the error message shown in Fig. 2 appears when the user initiate OptiInstrument application and the NI-VISA package is not installed.
- OptiInstrument 2.0 includes the option to install OptiInstrument samples during (or any time after) installation. The installation location for the samples folder can be defined (by default the samples folder will be installed in the user's **Documents**, under "OptiInstrument 2.0 Samples" folder.

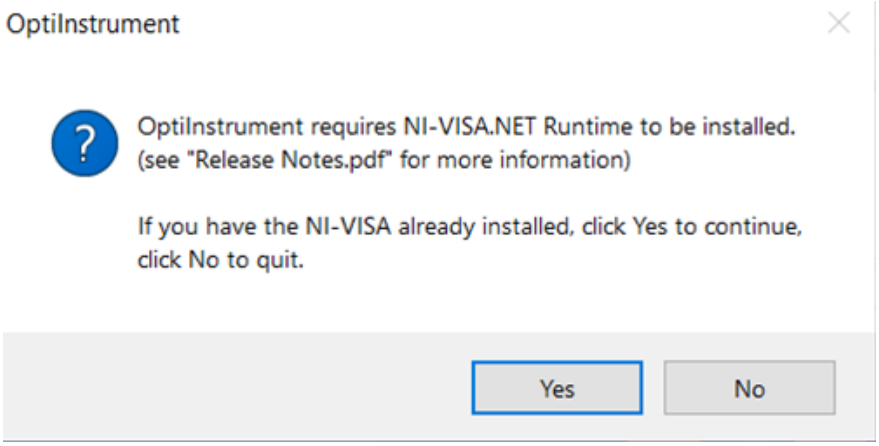

Fig. 1 OptiInstrument installation popup message

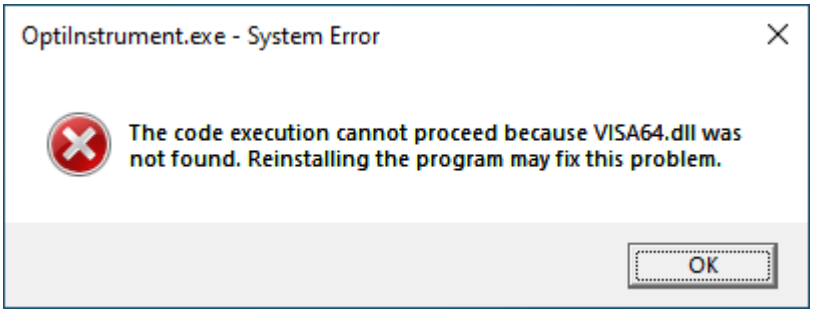

Fig. 2 OptiInstrument installation error message when NI-VISA Runtime is not preinstalled

Minimum Hardware and Software Requirements

- OptiInstrument requires the following minimum/recommended system configuration:
- Minimum PC configuration: PC with Pentium processor (E6, G Series) or equivalent.
- 8GB RAM.
- OptiInstrument requires the following third-party software packages to be installed:
- NI-VISA (NI-VISA.NET Runtime).
- Recommended PC configuration: PC with a clock speed > 2 GHz with 2-4 cores (e.g. Intel i3, i5, i7) and 16GB RAM or more.
- Operating Systems: Microsoft Windows 8.1/10 (**64-bit only!**)
- **Microsoft is shelving Windows 7;** we will not support Windows 7 starting this release. However, the software might run under Windows 7, but we do not guarantee it and we will not be able to provide technical support for bugs/crashes.
- 2 GB free hard disk space.
- 1280 x 1024 graphic resolution

Application Execution

- **Administrators:** when installing OptiInstrument for users with Restricted User Profile, install the sample files in a folder where these users have Read/Write access. By default, the sample files are installed in the current user's Document folder. OptiInstrument requires the read/write file access and will not work with read-only files.
- For the OptiInstrument Help feature to function properly, Adobe Acrobat Reader must be installed. To get the latest version please visit the Adobe website at http://www.adobe.com/.
- Some computers are configured in power saving mode to go to Hibernation or Sleep mode when they are not in use. It is recommended to disable this feature, especially when running unattended lengthy simulations. Typically, after the

simulation is complete, the computer idles and eventually goes to Hibernation. This causes the licensing platform drivers to invalidate the license. When the computer wakes up and resume its execution, OptiInstrument software will issue a message that the license is not available and terminate, losing the simulation results in the process. Please disable the computer hibernation feature to avoid this problem.

OptiInstrument Software Overview

The user-friendly graphical user interface (GUI) of **OptiInstrument 2.0** Software is shown in Fig. 3. It is a standalone tool that can be used to communicate and control different kinds of instruments. OptiInstrument uses the standard commands for programmable instruments (**SCPI**) to communicate **physically** or **remotely** with instruments. The tool uses standard communication interfaces such as **TCP/IP**, **USB**, **GPIB**, or a serial port (**RS232/RS485**). Users can load lists of SCPI commands from **XML files** or write individual commands to control the instrument(s). The commands appear in a **tree configuration**. A single command or a sequence of commands can be executed by OptiInstrument. A **Python script** can be generated for the SCPI commands, saved, loaded and executed by OptiInstrument or in a Python environment. OptiInstrument GUI has a built-in viewer and CSV file analysis window. The GUI supports dockable windows that can be split of the main GUI or placed anywhere in the GUI. OptiInstrument is ideal for automated testing and characterization.

OptiInstrument software version 2.0 does not require the user to install python software on his computer. A directory with required python and plotting directories are embedded and offered as part of the installer of OptiInstrument software.

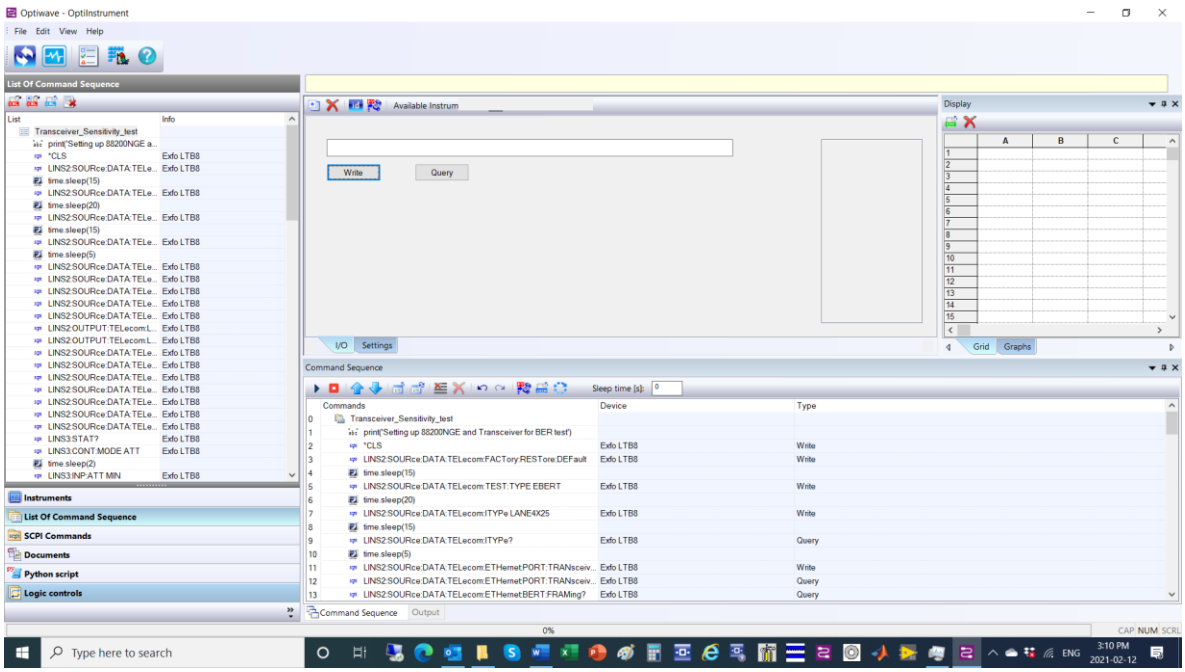

Fig. 3 OptiInstrument GUI

OptiInstrument

Key Features of OptiInstrument 2.0

- User friendly GUI for efficient and intelligent testing and characterization.
- Embedded python installation files.
- Support logic statement such as while (while loop), if, else, elif (else if), for (for loop).
- Independent delay time (sleep) for each individual SCPI command.
- Offer “Basic helpers” statements such as print, sleep, break from a loop, continue and variables to build complex SCPI command sequences.
- Error handling support for identifying the type of error in the sequence
- Addon GUI for data post processing, graphing and saving
- Save output results in a Jason file format.
- Live display of output results for each SCPI command in the sequence.

OptiInstrument Software Applications

- Remotely communicate with instruments
- Setup parameters of equipment
- Automate testing and characterization
- View generated signals
- Extract & save the data of generated signals for post processing
- Integrate instruments with photonics and systems simulation tools

Data Post Processing Addon GUI

OptiInstrument software Post Processing popup GUI is used for graphing, organizing and saving of the output data obtained from executing the sequence of commands. The post processing GUI is shown in Fig 4.

Fig. 4 OptiInstrument Post Processing GUI

OptiInstrument & Python Script

OptiInstrument software supports Python scripting. A Python script is generated for a single SCPI command or a list of commands using the tool. The generated script can be saved into a file. The generated script can be executed from OptiInstrument GUI or in **command prompt** or **Windows PowerShell**. A Python script can be loaded into OptiInstrument GUI and executed by the GUI. Fig. 5 shows a Python script generated for a sequence of SCPI commands and displayed in the GUI Python script pane. This capability allows users to execute features that are not supported by OptiInstrument GUI such as logic control and looping options.

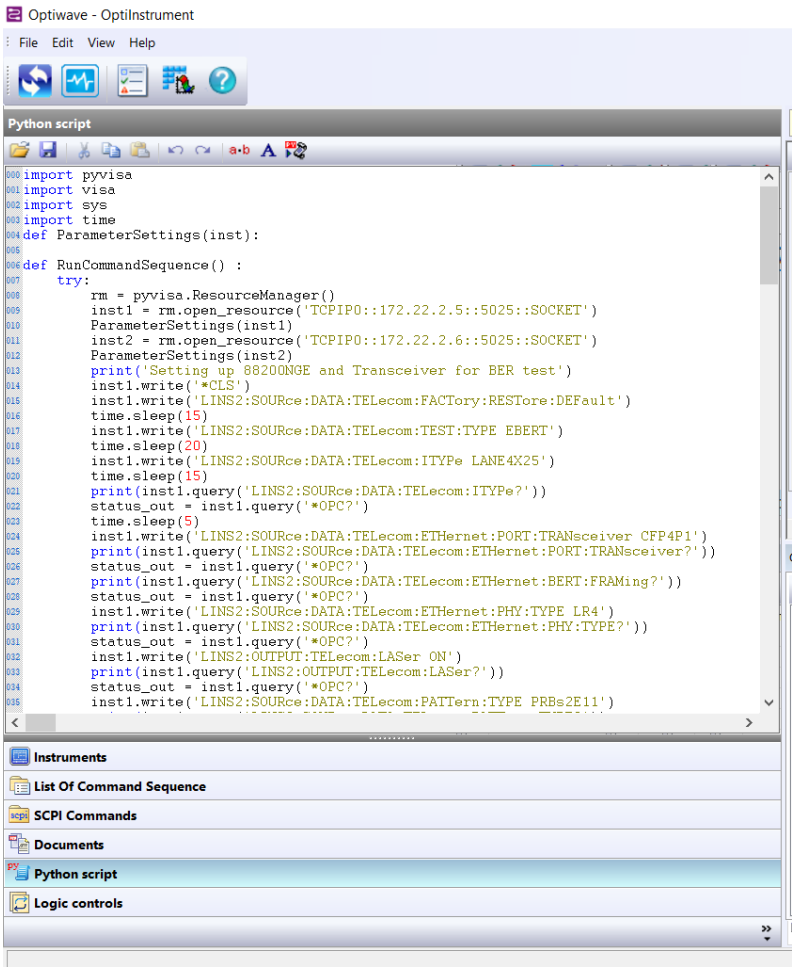

Fig. 5 Generated Python script for a sequence of SCPI command displayed in the Python script pane of OptiInstrument GUI

OptiInstrument

OptiInstrument 2.0 Example Library

OptiInstrument 2.0 Software has many examples that are created using commercial instruments from Rigol and EXFO. The examples are organized in subdirectories for each vendor. Each example has a readme file that describes the setup and the instrument(s)/card(s) used in each example as well as the result file(s). The **Samples** directory has also a subdirectory (**EXFO_General SCPI Commands**) for all SCPI command offered by EXFO for their different equipment. These commands are saved in XML files that can be loaded into OptiInstrument List of Command Sequence pane and used to build the desired SCPI command sequences.

1. EXFO Samples

a. CFP4 Longterm Sensitivity Test
b. CFP4 Transceiver Sensitivity Setup-I
c. CFP4 Transceiver Sensitivity Setup-II
d. CFP4 Transceiver Sensitivity Setup-III
e. EXFO OTDR card
f. Long Term Stability_LTB-8 cards
g. Double Nested Loops_LTB-8 cards
h. PowerBalzer_CFP4_EBERT
i. PowerMeter_VOA_CW Source
j. PowerMeter_2 CW Sources
k. Switch_OSA_2 CW Sources
l. Switch_OSA_VOA_4 CW Sources
m. Switch_OSA_VOA_CW Source

2. EXFO_General SCPI Commands

3. RIGOL Samples

a. AM waveform
b. Arbitrary waveform
c. Burt waveform
d. Harmonic waveform
e. PSK waveform
f. Pulse waveform
g. Ramp waveform
h. Sinewave
i. Square waveform