

---

# Keysight N7700 Photonic Application Suite

IL / PDL Measurement Engine

# Notices

© Keysight Technologies 2024

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Keysight Technologies as governed by United States and international copyright laws.

## Manual Part Number

N7700-91100

## Edition

Edition 15.0, September 2024

Printed in Germany

Keysight Technologies Deutschland GmbH  
Herrenberger Strasse 130,  
71034 Böblingen, Germany

## Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

## U.S. Government Rights

The Software is “commercial computer software,” as defined by Federal Acquisition Regulation (“FAR”) 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement

(“DFARS”) 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly, Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at <http://www.keysight.com/find/sweula>. The license set forth in the EULA represents

the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data.

## Warranty

THE MATERIAL CONTAINED IN THIS DOCUMENT IS PROVIDED “AS IS,” AND IS SUBJECT TO BEING CHANGED, WITHOUT NOTICE, IN FUTURE EDITIONS. FURTHER, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, KEYSIGHT DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED WITH REGARD TO THIS MANUAL AND ANY INFORMATION CONTAINED HEREIN, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PAR-

TICULAR PURPOSE. KEYSIGHT SHALL NOT BE LIABLE FOR ERRORS OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, USE, OR PERFORMANCE OF THIS DOCUMENT OR ANY INFORMATION CONTAINED HEREIN. SHOULD KEYSIGHT AND THE USER HAVE A SEPARATE WRITTEN AGREEMENT WITH WARRANTY TERMS COVERING THE MATERIAL IN THIS DOCUMENT THAT CONFLICT WITH THESE TERMS, THE WARRANTY TERMS IN THE SEPARATE AGREEMENT WILL CONTROL.

## Safety Notices

### CAUTION

A CAUTION notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a CAUTION notice until the indicated conditions are fully understood and met.

### WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

# Keysight Cybersecurity

## Product and Solution Cybersecurity

Keysight is dedicated to ensuring the cybersecurity of its products and solutions. For detailed information, visit [Product and Solution Cybersecurity](#).

## Report a Product Cybersecurity Issue

If you encounter a cybersecurity issue with a Keysight product, report it immediately at: [Product Cybersecurity Issue Reporting](#).

## Responsible Disclosure Program

Keysight encourages responsible disclosure of security vulnerabilities. For more details, visit [Responsible Disclosure Program](#).

## Product Software Updates

Keysight releases periodic software updates to fix known defects, incorporate product enhancements, and address cybersecurity vulnerabilities, if any. Ensure that your product software is always up to date. To search for software updates for your product, use the Photonic Applications Package Manager application. The Package Manager application could be downloaded from the Keysight Technical Support page at: <https://www.keysight.com/find/N7700>. It is also installed as part of the N7700 software installation. Press Win + S and type 'Photonic Applications Package Manager' in the Windows search bar. In the Package Manager application switch to the 'PAS' tab and check if software updates for your product are available.

## User Documentation

User documentation includes comprehensive information on features that may impact the secure deployment, use, and decommissioning of our products. This includes details on exposed network ports, the use of cryptography and authentication, and firmware security settings, if necessary.

# Compliance and Environmental Information

**Table 1 Compliance and Environmental Information**

Safety Symbol	Description
	<p>This product complies with WEEE Directive (2002/96/EC) marking requirements. The affixed label indicates that you must not discard this electrical/electronic product in domestic household waste.</p> <p>Product Category: With reference to the equipment types in WEEE Directive Annex I, this product is classed as a "Monitoring and Control instrumentation" product.</p> <p>Do not dispose in domestic household waste.</p> <p>To return unwanted products, contact your local Keysight office, or see <a href="http://about.keysight.com/en/companyinfo/environment/takeback.shtml">http://about.keysight.com/en/companyinfo/environment/takeback.shtml</a> for more information.</p>

# Contents

Keysight Cybersecurity 3

Compliance and Environmental Information 4

## 1 Quick Start Information

**Fast Multichannel IL / PDL Setup** 15

**Trigger Connections** 17

**General** 19

**Receiver Characterization** 20

**Receiver Characterization using B29xxA SMU instruments** 21

**Receiver Characterization using N7745A instruments** 22

**Software User Interface** 23

## 2 Getting Started

**About this manual** 26

**System Requirements** 27

**Software Installation** 28

**Running the IL/PDL Engine Configuration Wizard** 29

**Using the N7786B GPIB Interface (Remote GPIB)** 31

## 3 Fast IL / PDL Application

**General** 35

<b>Application: Fast IL / PDL</b>	36
Receiver Measurements	38
Application Setup Parameters	40
Instrument Setup Parameters	43
Advanced Parameters	47
Polarization Extinction Ratio Parameters	50
Performing a Reference Measurement	51
Managing References	58
Performing a Measurement	59
Details on PER and Dark Current Measurements	60
IL De-embedding	62
Static Mode	71
Re-Evaluating Measurements	77
Applying Plugins	78
Saving Measurements	78
Loading Measurements	79
Creating .agconfig Files from Measurement Files	79
Automatic History Saves	80
Exporting Measurement Data	85
Changing the Color Theme	85
Power/ Current Meter Zeroing	86
TLS Lambda Zeroing	86
Return Loss Calibration	87
Bias Settings	88
Configuration Handling	89
Basic Measurement Setup	89
Extended Measurement Setup	92

## 4 Automation

<b>COM Components</b>	94
<b>Creating a New Engine</b>	96
<b>Connecting to an Existing Engine</b>	98
<b>Performing a Measurement</b>	99

<b>Accessing the Measurement Result</b>	100
<b>Reference: Interface “IEngineMgr”</b>	102
property Version	102
method IsVersionGreaterOrEqual	102
method NewEngine	102
method OpenEngine	103
property EngineIDs	103
method DeleteEngine	103

<b>Reference: Interface “IEngine”</b>	104
property Version	104
method IsVersionGreaterOrEqual	104
method Activate	104
method DeActivate	105
property Active	105
property Busy	105
property EmulationMode	106
method LoadConfiguration	106
property Configuration	106
method StartReference	106
method StartCustomReference/ StartCustomReferenceSinglePort	107
method ClearReferenceChannels / ClearReferenceChannelsSinglePort	107
method CopyReferenceChannel / CopyReferenceChannelSinglePort	108
method UndoReferenceOperation	108
property ReferenceList	109
method DeleteReferenceFile	109
method ValidateSettings	109
method ValidateSettingsNoRefCheck	109
property DarkIncludeMeasurement	110
property ERIncludeMeasurement	110
property HighCapacitanceMode	110
method StartMeasurement	111
method StartMeasurementRepeat	111
method StopMeasurement	111
method FileSave	112
property MeasurementResult	112
property ResolutionValues	112
property Resolution	113
method ReCalculate	113
property MaxAllowedRefLoss	113
property PWMCount	113
method GetPWMChannelCount	114

property TLSCount	114
method GetTLSInterface	114
property KeepRawData	115
property PWMAutoRanging	115
property MPPMAutoGain	115
property TLSExtendedSweepRange	116
property GenerateTETMData	116
property GenerateReturnLossData	116
property ReturnLossModulePresent	117
property GenerateResponsivityData	117
property GenerateDiodeCurrentData	117
property ElectricalInputsPresent	117
method IsBiasEnabled	118
method GetBias	118
method SetBias	118
property BiasMode	119
method ZeroAllMPPMs	119
property LambdaZeroingMode	119
method ZeroAllTLS	120
method CalibrateRLM	120
property RLMCalReflectorValue	120
property EventPropertiesChanged	121
property EventMeasurementFinished	121
property EventreferenceOperationFinished	121
property ProtocolText	121
property ProtocolMin	122
property ProtocolMax	122
method GetProtocolTextAt	122
property UserInputWaiting	122
property UserInputPrompt	123
property UserInputChoice	123
method UserInputResponse	123
method DataLoggingTimeout	124
property ImmediateEvaluation	124
property QuickPolarimeterSetup	124

property MeasurementSpeedLegacyMode	125
property SOPSearchQuickMode	125
method ForceMediumN7786BInitialization	125
method ForceFullN7786BInitialization	126
method ForceTLSInitialization	126
method StartILDeembeddingMasterPortMeasurement	126
method StartILDeembeddingSpecificPathMeasurement	126
method GetILDeembeddingFileList	127
method SetILDeembeddingFileList	127
property ILDeembeddingTargetFolder	128
property EnableILDeembedding	128
method SetStatic	129
method StopStabilizer	129
property StaticWavelength	129
property StaticTLSPower	130
property StaticPWMPort	130
property StaticPolarizationMode	130
property StaticPolarizationMaxMinMode	131
property StaticPolarizationPSPMode	132
property StaticFilename	132
property StaticStokesParameters	133
property StaticWaveplateOrientations	133
method ReadCurrentStokesParameters	134
property ReadCurrentWaveplateOrientations	134
method ReadCurrentStabilizerState	134

<b>Reference: Interface “IEngineTLS”</b>	135
property ModelCode	135
property Bands	135
property Include	135
property WavelengthStart	136
property WavelengthStop	136
property SweepRates	136
property SweepRate	136
property PWMSensitivity	137
property PWMSensitivityRef	137
property RLMSensitivity	137
property RLMSensitivityMon	138
property OutputPorts	138
property TLSOutputPort	138
property TLSPower	139
property OutputActive	139
property UseLambdaLog	139
method LoadReference	140
method SaveReference	140
property ReferenceDescription	140
method ReferenceAvailableAtChannel	141
method ReferenceAveragePowerAtChannel	141
property ERWavelengthStart	141
property ERWavelengthStop	142
property ERUseSweepRange	142
property ERWavelengthSamples	142
property ERTLSPower	143
property ERPWMSensitivity	143
<b>Alphabetical Automation Index</b>	144

## 5 Troubleshooting

<b>Symptoms and Solutions</b>	148
-------------------------------	-----



# 1

## Quick Start Information

[Fast Multichannel IL / PDL Setup](#) / 15

[Trigger Connections](#) / 17

[General](#) / 19

[Receiver Characterization](#) / 20

[Receiver Characterization using B29xxA SMU instruments](#) / 21

[Receiver Characterization using N7745A instruments](#) / 22

[Software User Interface](#) / 23

This section illustrates how to connect your instruments optically and electrically for typical applications. It is intended as a quick start guide, although it is strongly recommended that you read through Chapter , [“Getting Started,”](#) on page 25 and Chapter , [“Automation,”](#) on page 93 of this User's Guide and through the user's guides of all instruments prior to operating the instruments.

### NOTE

You need to be logged in as an administrator to install the *Photonic Application Software Suite*.

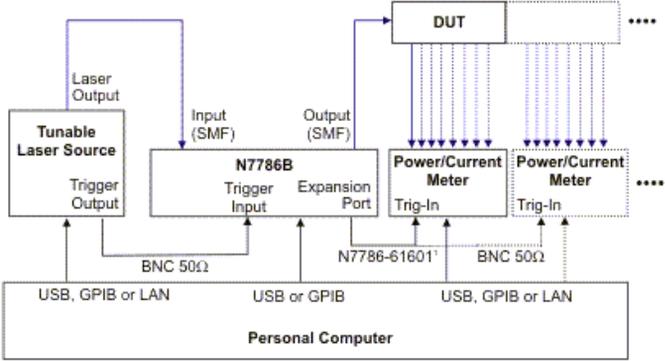
The instrument drivers should be installed on the PC before connecting the instruments, especially via USB. This can be ensured by first installing the indicated packages of the Photonic Application Suite. If you have connected the instrument(s) prior to software installation, you may have to delete the instrument(s) manually from the Windows Device Manager. This does not apply to instruments that are connected via GPIB.

After installation of the Photonic Application Suite, connect the instruments to the local area network (LAN), USB or GPIB. You may use USB hubs to increase the number of USB ports on your PC. Certain instruments can also be accessed through your local network. For the connected instruments to be found by Photonic Application Suite engines, they need to be configured in VISA, for example with the Connection Expert which is part of the IO Libraries Suite, if Keysight VISA is primary.

After the instruments have been connected (and recognized in the case of USB instruments), start the *Launch Pad* located in the start menu. Then click on "Insertion Loss / Polarization Dependent Loss". On first start, the *Configuration Wizard* comes up and searches for all suitable instruments. Follow the instructions given by the *Configuration Wizard*. After finishing the wizard, the application comes up with all configured instruments. If you change your hardware configuration, run the wizard again by clicking on *File->Run Configuration Wizard*.

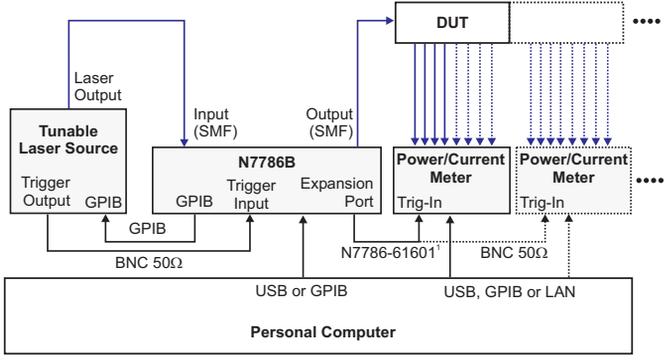
# Fast Multichannel IL / PDL Setup

The IL/PDL engine can be used in quite a large number of different hardware configurations. Shown here are two rather common, basic setups, followed by an extended configuration with three TLS, an optical switch and return loss module. After that, important details on the trigger setup and receiver characterization are explained.



<sup>1</sup>: N7786-61601 is color coded with red heat shrink tubing at both ends

Figure 1 Fast Multichannel IL / PDL Setup



<sup>1</sup>: N7786-61601 is color coded with red heat shrink tubing at both ends

Figure 2 Fast Multichannel IL / PDL Setup (Remote GPIB)

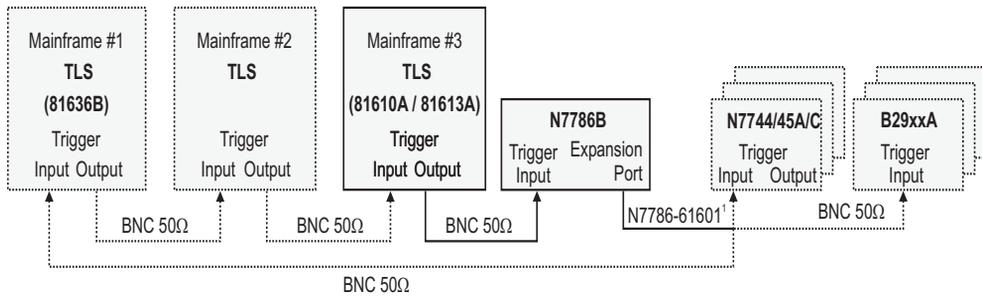
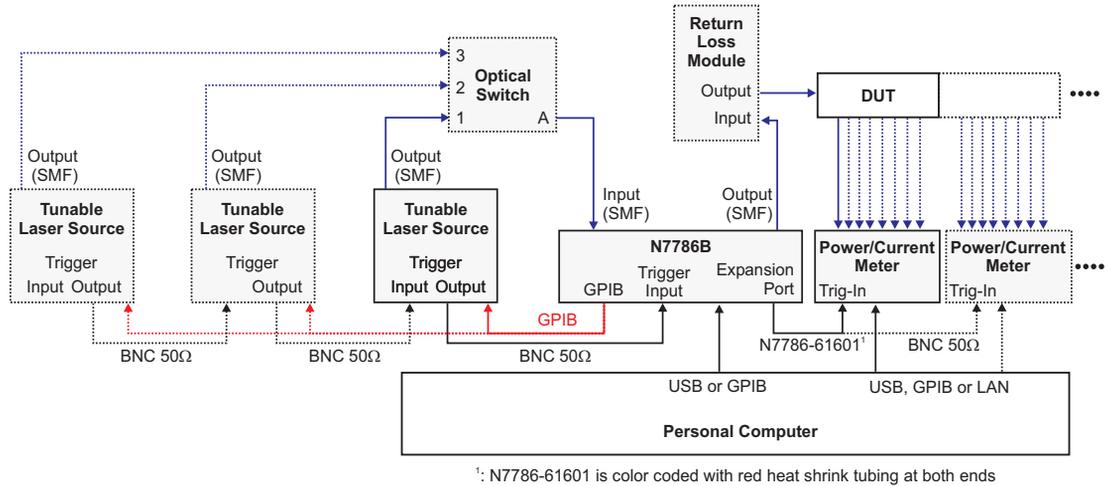


Figure 3 Fast Multichannel IL / PDL Full-Blown Setup and Trigger Details

## Trigger Connections

The following points must be kept in mind while configuring the triggering connections for setups with multiple instruments.

- The engine automatically configures all mainframes and laser instruments that are included in the current configuration, assuming they are daisy-chained as shown in the above illustration. If an instrument in the physical trigger path is not included in the current configuration, it will not be configured by the engine and thus, most likely, interrupt the daisy chain.  
When other instruments not used by the software are connected to the trigger paths, care should be taken to avoid unexpected behavior.
- Connect each trigger output of the TLS mainframes to the trigger input of the next mainframe and the last trigger output to the trigger input of the N7786B.
- If an 81610A or an 81613A return loss module is used in the setup, it must be placed in the last mainframe of the TLS/mainframe daisy chain, i.e., the one that is connected to the trigger input of the N7786B so that it can receive triggers from any of the TLS. (This is automatically fulfilled if only a single mainframe is used in the setup).
- If one or more 81636B power meters are used in a mainframe with one of the TLS, they must be placed in the first mainframe of the TLS/mainframe daisy chain (This is automatically fulfilled if only a single mainframe is used in the setup). The input trigger of this mainframe should be connected to the trigger output of the N7786B.
- Use RG-58 (50 Ohm) BNC cables to connect the mainframes to one another and the last mainframe of the daisy chain to the N7786B trigger input.
- Use the N7786-61601 trigger cable to connect the N7786B expansion port to the first power/current meter instrument.
- If only N7744A/N7745A or N7744C/N7745C power/current meters are used, their trigger connections can be daisy chained as well. They will automatically be configured such that input triggers are routed to the trigger outputs as well.
- If more than one power/current meter is used in the setup that is not an N7744A/N7745A or N7744C/N7745C, use BNC Tee connectors to split the trigger signal accordingly.

- For measuring photocurrent with the B2900-series source/measure units, use the N1294A-031 trigger adapter to connect the power meter / current meter trigger signal (i.e., the output of the N7786-61601 trigger cable or a respective tee connector) to the Digital I/O port of each SMU. (DIO9).  
There is currently no trigger passthrough configuration for B29xxA instruments.



Figure 4 N7786-61601 Trigger Cable



Figure 5 N1294A-031 GPIO - BNC Trigger Adapter

## General

**NOTE**

When using 81636B and or B29xxA instruments, the maximum number of data points will be less than when using N7744A/N7745A or N7744C/N7745C instruments, hence limiting the wavelength range at high resolutions. As long as such instruments are configured for the current setup, this limitation applies to any N7744A/N7745A or N7744C/N7745C in the same setup as well. If necessary, the engine will limit the sweep range and inform the user accordingly.

---

**NOTE**

If using any of the following, the optical switch, the return loss module and 81636B power meter modules can be placed in the mainframe of one of the tunable laser sources. However, special restrictions regarding the trigger configuration might apply (see [Trigger Connections](#) on page 17).

---

**NOTE**

In case of multiple TLS, connect the optical TLS outputs to ports 1 to 4 of the optical switch. You don't have to connect specific lasers to specific ports of the optical switch. The engine will perform a port auto-detection upon activation. Connect switch port A to the optical input port of the N7786B.

---

**NOTE**

Refer to [Basic Measurement Setup](#) on page 89 for more information on the measurement setup.

---

## Receiver Characterization

For measuring a component with integrated photodetectors, like a receiver optical subassembly (ROSA), the output photocurrent can be detected to determine the dependence of responsivity on the input wavelength and polarization.

Measurements are usually done by connecting the photodiode bias pins of the receiver device to the current measurement instruments (N7745A-E01/-E02/B2901A/B2902A/B2911A/B2912A).

For characterizing receivers with a significant capacitance (e.g. 100nF) in parallel to the bias contacts, as is commonly the case, to use the High Capacitance Mode (see [Instrument Setup Parameters](#) on page 43). This uses slower sampling to accommodate the lower analog bandwidth of the bias circuit.

Please note that in High Capacitance Mode, only current meter ranges of 0dBm or above are supported with the B2900-series SMU.

### NOTE

When doing receiver characterization, power / current meter ranges will still be shown as dBm. 10/0/-10/-20dBm correspond to 10/1/0.1/0.01mA range.

---

## Receiver Characterization using B29xxA SMU instruments



Figure 6 B29xxA Front View

The following points should be kept in mind when performing receiver characterization using B29xxA SMU instruments:

- Use the center connector pair (Force High (red) and Force Low (black)) of each port of the B29xxA instruments for receiver measurements.
- Positive bias voltages set in the engine configuration will cause a bias voltage with the high potential at the red connector interface.
- Care should be taken regarding electromagnetic interference. Wire loops should be avoided. Best results are achieved with shielded cables. Twisted wires and wires within a single flat ribbon cables usually work fine as well.
- B2902A and B2912A instruments have a second measurement port at the rear panel.
- The measurement ports of B29xxA instruments are set to floating, i.e., they don't have an internal connection to case ground / protective earth. This allows for maximum flexibility regarding DUT configurations.
- B29xxA instruments are configured to limit measured currents to 7mA.
- At bias voltages with a magnitude of 2V or more, B29xxA instruments will operate the source in a different voltage range and thus add some noise to the measurement results. Best results will be achieved for bias voltages below 2V.

Using the High Capacitance Mode of the measurement engine yields better results in case of noisy signals.

- When any B29xxA instrument is part of the active setup, power / current meter ranges will be limited to 0dBm and above.

## Receiver Characterization using N7745A instruments

A special customization of the N7745A, identified by option E02 replaces 4 of the optical power meter ports with photocurrent input ports through triaxial front-panel connectors.

For connecting the DUT, insulated triax-to-coax adapters (1250-2650) are recommended. The (negative) bias voltage (-4 to 0V) will then be connected to the coax shielding. While the instrument is powered, the center contact will be at N7745A case ground potential.

N7745A instruments measure photocurrent in a single polarity only. The photodiode cathode must be connected to the triax/coax center contact. Only negative biasing is available.

A single N7745A port's bias voltage source can be used for multiple measurement ports. This is required for DUTs with multiple photodiodes with common anodes.

N7745A does not have a controlled current limitation. However, there is an absolute limitation, since their bias voltage sources can only provide a maximum current of 20mA.

Care should be taken regarding induced distortions. Best results are achieved with shielded cables. Twisted wires and wires within a single flat ribbon cables usually work fine as well.

## Software User Interface

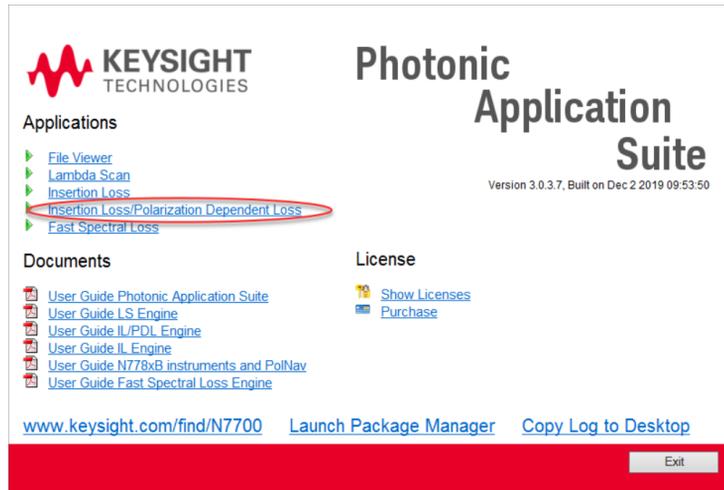
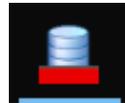


Figure 7 The *Launch Pad* gives you access to the installed components. Click on “Insertion Loss / Polarization Dependent Loss” to start the IL / PDL Engine

## NOTE

The IL / PDL Engine consists of a *Client Software* which handles the graphical input/output and a *Server Software* which handles the communication with the hardware. For remote control or automation, this knowledge might be useful since the controlling program communicates directly with the *Server Software*. The server will be automatically started when you click on “Insertion Loss / Polarization Dependent Loss”. The presence of the server is indicated in the task bar:





# 2 Getting Started

[About this manual](#) / 26

[System Requirements](#) / 27

[Software Installation](#) / 28

[Running the IL/PDL Engine Configuration Wizard](#) / 29

[Using the N7786B GPIB Interface \(Remote GPIB\)](#) / 31

This chapter gives information for setting up your instruments.

## About this manual

This manual covers the single-sweep IL / PDL application that uses an N7786B Polarization Synthesizer with one to three tunable laser sources and a number of power / current meter instruments and optionally an 81610A or 81613A return loss module.

You might also refer to the *N7700 Keysight Photonic Application Suite User's Guide* to get information on the core Photonic Application Suite functionality including the File Viewer, Plugins, and the general COM API.

## System Requirements

- One to three Keysight continuous-sweep tunable laser source (N7776C, N7778C, 81960A, 81940A, 81980A or any TLS models for 8164B Slot 0, e.g., 81600B, 81606A, 81607A, or 81608A)
- One Optical Switch (81595B, N7731A, or N7734A) in case more than one tunable laser source is used
- One N7786B Polarization Synthesizer
- One or more power / current meter instruments:
  - N7744A /N7745A
  - N7744C / N7745C
  - N7745A-E01/-E02
  - B2901A / B2902A / B2911A / B2912A
  - 81636B
- Up to one 81610A / 81613A return loss module
- One or more 8163B or 8164B mainframes
- One Single-mode fiber patchcord from laser to N7786B
- Fiber patchcord for reference and DUT configurations
- One N7786-61601 Trigger Cable
- N7786-61602 Trigger Cable(s) (one for each B29xxA)
- One or more BNC trigger cables (RG58, 50 Ohms)
- LAN, GPIB or USB cables
- Personal Computer:
  - Operating System: Microsoft Windows 10 (64 bit)
  - Depending on the instruments, PC interfaces to LAN, USB, and GPIB can be used.

## Software Installation

The IL/PDL engine communicates with the instruments over a VISA software layer that should be installed before connecting the instruments. Keysight VISA is provided by installing the IO Libraries Suite, which also provides drivers that may be required by some devices, like GPIB interface adapters and instruments connected via USB. Consult the instrument documentation for establishing connections with VISA.

All drivers and supplements necessary for using instruments with the IL/PDL engine are included in the Photonic Application Suite installation packages. The Package Manager will indicate if any required packages are missing and still need to be installed.

Refer to the *N7700 Photonic Application User's Guide (N7700A-91001)* for information on how to install this software.

Instruments connected via GPIB and often USB will usually be recognized automatically for VISA, for example, by Connection Expert when using Keysight VISA. Instruments connected by LAN might need to be added manually. Note that the N77xxC family of instrument will also be identified as LAN instruments in VISA and do not require extra USB drivers. For use with the IL/PDL engine, instruments should be connected with the primary VISA if more than one VISA installation is used.

### NOTE

**You need to be logged in as an administrator to install the *Photonic Application Software Suite*.**

---

## Running the IL/PDL Engine Configuration Wizard

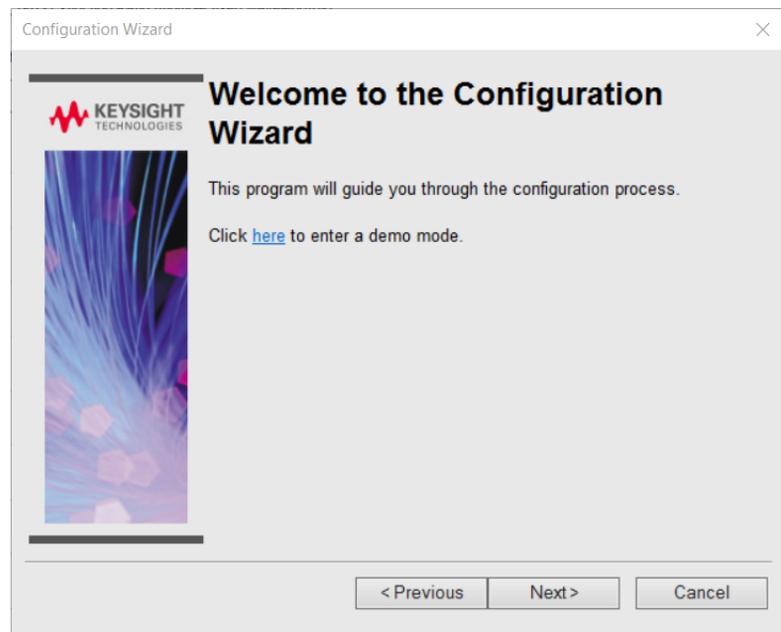
### NOTE

If Keysight VISA is used, all instruments need to be identified using the Keysight Connection Expert before the Photonic Application Suite Configuration Wizard is able to add them to the list of available devices. The Connection Expert is part of the Keysight IO-Libraries. Start the Connection Expert and click the Refresh All button. When all connected instruments have been identified proceed with the steps below.

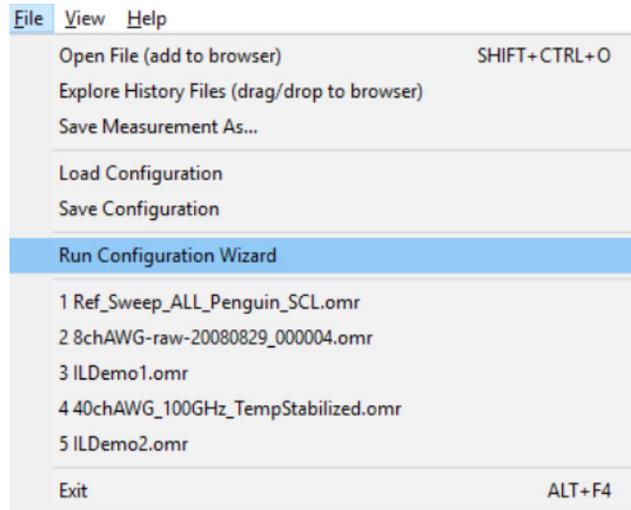
This procedure is only required if a certain supported instrument is connected to the PC for the first time.

Now that the software is installed and the instrument is connected and turned on, you can start the **IL / PDL Engine** and run the Configuration Wizard.

On first startup the Configuration Wizard will come up automatically.



If you want to rerun the Configuration Wizard, for example to use a different set of instruments later on, you can select Run Configuration Wizard from the File menu.



Please follow the instructions shown by the Configuration Wizard.

## Using the N7786B GPIB Interface (Remote GPIB)

The N7786B instrument can be used as GPIB-Master to control tunable laser sources using GPIB. This mode is referred to as Remote GPIB. In that case, the N7786B has to be connected to the PC via the USB port and the tunable laser source has to be connected to the N7786B with a GPIB cable.

**NOTE**

The IL / PDL engine of the Photonic Application Suite fully supports the remote GPIB port for tunable laser sources. However, it does not behave like a standard GPIB interface for other software such as LabVIEW™.

---

**NOTE**

The Remote GPIB mode cannot be used to connect N7744A / N7745A / B29xxA instruments to the N7786B. Connect such instruments directly to the PC, e.g. via the USB port.

---



# 3 Fast IL / PDL Application

General	/ 35
Application: Fast IL / PDL	/ 36
Receiver Measurements	/ 38
Application Setup Parameters	/ 40
Instrument Setup Parameters	/ 43
Advanced Parameters	/ 47
Polarization Extinction Ratio Parameters	/ 50
Performing a Reference Measurement	/ 51
Managing References	/ 58
Performing a Measurement	/ 59
Details on PER and Dark Current Measurements	/ 60
IL De-embedding	/ 62
Static Mode	/ 71
Re-Evaluating Measurements	/ 77
Applying Plugins	/ 78
Saving Measurements	/ 78
Loading Measurements	/ 79
Creating .agconfig Files from Measurement Files	/ 79
Automatic History Saves	/ 80
Exporting Measurement Data	/ 85
Changing the Color Theme	/ 85
Power/ Current Meter Zeroing	/ 86
TLS Lambda Zeroing	/ 86
Return Loss Calibration	/ 87
Bias Settings	/ 88

Configuration Handling / 89

Basic Measurement Setup / 89

Extended Measurement Setup / 92

## General

You can run the various applications of the Photonic Application Suite from the Launch Pad that is found in the Start menu.



Figure 8 The *Launch Pad* gives you access to the installed components. Click on “Insertion Loss / Polarization Dependent Loss” to start the IL / PDL Engine.

## Application: Fast IL / PDL

Use this application for fast wavelength-resolved multi-channel IL / PDL measurements. It uses a continuous sweep of the tunable laser source (TLS) to determine the following parameters versus wavelength:

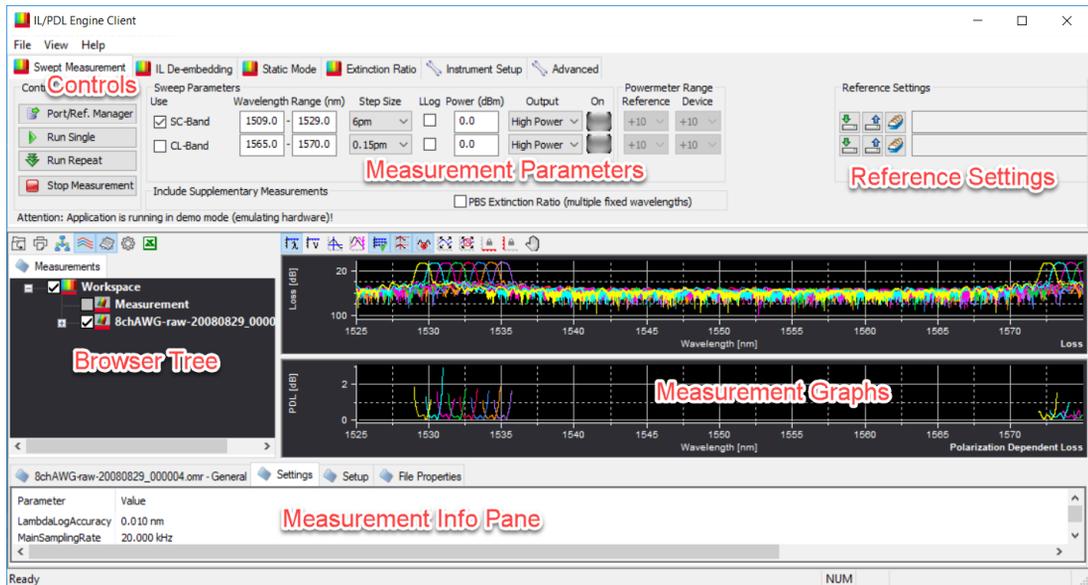
- PDL
- Power/Loss
- TE / TM
- Mueller data
- Return Loss
- Photodiode Responsivity (average and max/min)
- Photodiode Current
- Common Mode Rejection Ratio (DC)
- Polarization Extinction ratio (fixed wavelengths)
- Photodiode Dark Current (TLS output turned off)

See [Fast Multichannel IL / PDL Setup](#) on page 15 and [Basic Measurement Setup](#) on page 89 for details on how to connect the DUT and the trigger cables.

When working with a more complex measurement setup, including optical switches and splitter, for instance, these can be characterized once, then de-embedded from sweep measurement results later on. See [IL De-embedding](#) on page 62 for details.

When working with highly polarization-dependent devices, it might be desired to perform additional custom measurement operations, once input polarization has been aligned with either of the DUT's main axes. See [Static Mode](#) on page 71 for details on how to do this.

Before starting a measurement you should make sure that the measurement parameters suit your measurement task.



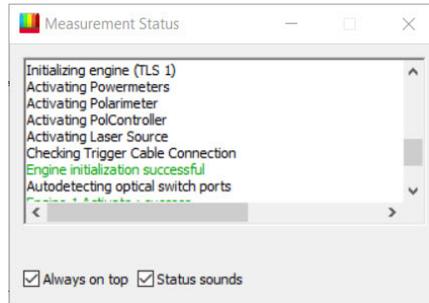
The measurement control buttons are displayed in the top left portion of the application window, the measurement parameters are shown to the right of these and the reference properties are shown even further to the right. In case the reference descriptions does not fit on your screen, you can see the full line in the tooltip by hovering the mouse pointer over the reference description control. Details of the measurement parameters are given in the tables in the next section.

The browser tree to the left of the application window lists the current measurement and all open files and allows configuring, which measurements, which traces (e.g. IL, PDL, TE/TM) and which channels are displayed in the measurement view to the right. The measurement information pane can be toggled by pressing the Show/Hide Info Pane icon above the browser tree.

For information on how to configure the Measurement File Viewer area, i.e. the browser tree and the display of the measurement traces, refer to the *Photonic Application Suite User's Guide*.

Information on the current operation progress is displayed in the *Measurement Status* window. Sometimes user inputs are required, e.g. choosing whether to continue or to abort a certain operation. These user input requests are shown in the *Measurement Status* window. There will be then different response buttons at the bottom of the status window.

Enable the *Status sounds* check box to receive auditory notifications (beeps) when the status of the engine operations change, for example, when a measurement completes or when an error is encountered.



## NOTE

If the *Measurement Status* window is closed, it will open automatically, once any user input is required. It can be opened manually by checking *Measurement Status* in the *View* menu.

If the *Measurement Status* window is not visible at all, it may have been moved to a screen region that is currently not visible. By choosing "Reset Status Window Position" from the "View" menu, the *Measurement Status* window will be put to the top left corner of the application window.

## Receiver Measurements

The IL/PDL engine allows for characterizing wavelength dependent properties of passive optical components as well as O/E receiver devices. For measuring receiver devices, an appropriate instrument for measuring photocurrent is required. Such instruments are Keysight N7745A-E01 or N7745A-E02 specials, as well as Keysight B2901A/B2902A/B2911A/B2912A Precision Source / Measure Units.

Certain aspects have to be considered when using the IL/PDL engine for receiver measurements.

### Bias Voltages

The IL/PDL engine performs measurements at the bias voltage contacts of the receiver device. The most important consideration is the bias voltage polarity. While N7745A-E01 and N7745A-E02 instruments only allow for a negative bias voltage, B29xxA instruments can be used for either bias voltage polarity. For single port devices, this should not mean any

limitation, as long as one pays attention to the orientation of the DUT. For multiport devices that share a common contact, such as common anode or common cathode configuration, this may require special considerations. Refer to section [Extended Measurement Setup](#) on page 92 for further details on configuring the setup and section [Instrument Setup Parameters](#) on page 43 for details on bias settings.

### **High Capacitance Mode**

Receiver modules often have a capacitance in parallel to the bias voltage contacts. Since the IL/PDL engine is switching the state of polarization at several kHz, this may impact the measured response and usually leads to reduced PDL / PER values. If the DUT has a significant capacitance in parallel to the bias connectors (e.g. 100nF), we recommend using the High Capacitance Mode (see [Instrument Setup Parameters](#) on page 43). Using this mode, the measurements will be performed at a reduced sampling rate and power ranges will be restricted to 0dBm (1mA) and above. At the same time, averaging time is significantly increased, which can reduce noise and extend the dynamic range.

### **Reference Measurements**

Receiver measurements require an optical reference measurement, so an N7744A/N7745A port, N7744C/N7745C port, or an 81636B power sensor is required for reference measurements.

### **Zeroing**

No matter whether characterizing passive optical components or receiver devices, best performance is acquired if zeroing is performed for the power / current meter instruments, once they have settled at operating temperature. For optical measurement ports, zeroing is performed after the detector has been covered. For electrical measurement ports, DUT and connecting cables must be removed prior to starting the zeroing operation.

**NOTE**

The IL/PDL engine will use the measured responsivity data to compute the Common Mode Rejection Ratio (DC) by evaluating every pair of neighboring ports of the measurement results.

These don't necessarily have to be physically neighboring ports of the current meter instruments. Refer to [Performing a Reference Measurement](#) on page 51 for details on how to obtain and handle reference measurements. After taking a reference measurement on an optical port, you have to copy the reference data to all electrical ports to be used. It is recommended to clear the reference data on the optical port then, since the optical port is not connected during the receiver measurement and would show only noise.

For the CMRR results to be valid, ensure that the DUT's balanced photodiode pairs are connected to port pairs according to the reference obtained before.

---

#### Application Setup Parameters

The application setup parameters are grouped in separate tabs.

The Measurement Setup parameters are located on the Swept Measurement tab:

Include Supplementary Measurements  Dark Current  PBS Extinction Ratio (multiple fixed wavelengths)

Sweep Parameters

Use	Wavelength Range (nm)	Step Size	LLog	Power (dBm)	Output	On	Powermeter Range Reference	Powermeter Range Device
<input checked="" type="checkbox"/> SC-Band	1509.0 - 1529.0	6pm	<input type="checkbox"/>	0.0	High Power	<input type="checkbox"/>	+10	+10
<input type="checkbox"/> CL-Band	1565.0 - 1570.0	0.15pm	<input type="checkbox"/>	0.0	High Power	<input type="checkbox"/>	+10	+10

Return Loss Range  
Return  Monitor

References

<input type="button" value="←"/> <input type="button" value="→"/> <input type="button" value="🗑"/>	1456 nm - 1582 nm @ 20 nm/s (regular)
<input type="button" value="←"/> <input type="button" value="→"/> <input type="button" value="🗑"/>	1509 nm - 1641 nm @ 20 nm/s (regular)

The parameters and buttons are explained in the table below.

**Table 2** Application Fast IL/PDL Measurement Parameters (per TLS)

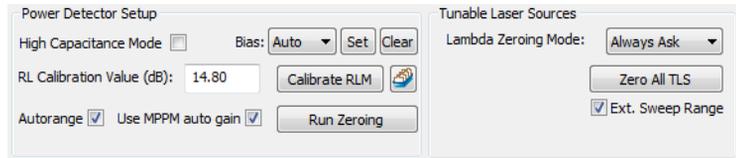
Parameter	Description
Dark Current:	If checked, a dark current measurement will be performed before the actual measurement. This is intended for use with current measurement instruments such as N7745A-E01/-E02 specials or B29xxA instruments.
PBS Extinction Ratio:	If checked, PER measurements will be performed at fixed wavelengths before the actual sweep measurement. Refer to sections "Polarization Extinction Ratio Parameters" on page -50 and "Details on PER and Dark Current Measurements" on page -60 for further details.
Use	Defines whether the TLS should be included in subsequent measurement operations.
Start wavelength (nm):	Defines the start of the wavelength sweep.

Parameter	Description
Stop wavelength (nm):	Defines the stop of the wavelength sweep.
Step size (pm):	Defines the nominal step size and thus the sweep rate of the laser. Note, that the number of data points changes with the step size since the sampling rate is constant.
Lambda Logging:	Keysight tunable laser sources provide a built-in optical wavemeter to increase wavelength accuracy. Check this box to use the internal wavemeter functionality (recommended).
TLS Power (dBm):	Defines the optical output power of the laser source.
Output:	If the TLS supports more than one optical output, the output is chosen here. The choices are "High Power", "Low SSE", "Both HR" and "Both LR". When selecting "Both HR", both outputs are active with the high power output being regulated. When selecting "Both LR", both outputs are active with the Low SSE output being regulated.
On:	This button shows the output state of the TLS and can be used to manually toggle the output state. Please note that it will not be updated if the TLS output state is changed at the TLS front panel.
Powermeter Range - Reference (dBm):	Sets up the power range for the reference measurement. The value defines the maximum allowed power to the power meter. Larger powers will result in clipped results and cause the reference measurement to fail. In case of reduced power at the power meter (e.g. intermediate instrumentation with significant loss) the range should be reduced accordingly.

Parameter	Description
Powermeter Range - Device (dBm):	<p>Sets up the power range for the measurement. The value defines the maximum allowed power to the power / current meter. Larger powers will result in clipped results. In case of reduced power at the power / current meter (e.g. DUTs with significant loss) the range should be reduced to utilize maximum dynamic range.</p> <p>Note: When using current meters, +10/0/-10/-20dBm correspond to 10/1/0.1/0.01mA ranges.</p> <p>Note: When <b>High Capacitance Mode</b> is used or a B29xxA Precision Source / Measure Unit is part of the current configuration, Powermeter Range will only allow values of 0dBm or above.</p>
Return Loss Range - Return (dBm):	<p>Sets up the power range of the return loss module (if used). The value defines the maximum allowed power returned to the detector. Larger powers will result in clipped results. In case of low power at the return loss module (e.g. DUTs with high return loss) the return loss module range should be reduced to utilize maximum dynamic range.</p>
Return Loss Range - Monitor (dBm):	<p>Sets up the power range of the monitor path in the return loss module (if used). The value defines the maximum allowed power. Larger powers will result in clipped results. In case of low power at the input of the return loss module the range should be reduced to utilize maximum dynamic range.</p>
References:	<p>This section contains buttons for reference handling (refer to <a href="#">Managing References</a> on page 58 for details). It also shows a comprehensive description of the reference currently selected for the TLS.</p>

## Instrument Setup Parameters

The Instrument Setup parameters are located on the Instrument Setup tab:



## NOTE

: The property **PWM Channels** or formerly **NumberOfChannelsToStore** has been removed in release 1.4. To choose which power / current meter ports should be evaluated, use the **Port/Reference Manager** to either provide or remove reference data from the respective ports. Only ports with reference data available will appear in the measurement result data.

For details on Bias, RLM Calibration and Zeroing, see sections [Bias Settings](#) on page 88, [Return Loss Calibration](#) on page 87, [Power/ Current Meter Zeroing](#) on page 86 and [TLS Lambda Zeroing](#) on page 86.

A brief description of the instrument setup parameters is shown in the table below.

**Table 3 Application Fast IL/PDL Instrument Setup Parameters**

Parameter	Description
High Capacitance Mode:	<p>If checked, measurements will be performed at a reduced sampling rate, allowing for electrical devices with significant parallel capacitances, such as receiver modules. See “Receiver Measurements” on page -38 for further details.</p> <p>High Capacitance Mode can also be helpful when characterizing passive optical components with very high PDL.</p>
Autorange:	<p>If turned on, the power / current meter (and return loss module) ranges (during regular and reference measurements) are adjusted automatically for each sweep.</p> <p>The range does not change during the sweep.</p>
Ext. Sweep Range	<p>If checked, power meter sampling will start sooner after the TLS sweep start. This increases the available sweep range, especially when using high sweep speeds, or operating close to the minimum allowed TLS wavelength. Measurement data in the wavelength region extended this way may show slightly reduced performance.</p> <p>Please note that reference measurements taken with this property set to true can only be used if the current measurement setting is also set to true and vice versa.</p> <p>References taken with extended sweep range active will be indicated in the GUI by an asterisk (*) after the wavelength range.</p> <p>The default setting is off.</p>
RL Calibration Value (dB):	<p>This is the return loss value of the reference reflection artifact used in the return loss calibration procedure (refer to “Return Loss Calibration” on page -87 for details).</p>

Parameter	Description
Lambda Zeroing Mode:	<p>There are a number of reasons (described below) that may require a lambda zeroing operation. Such an operation takes several ten seconds to complete. To minimize the impact on running measurements, but keeping up the wavelength accuracy, choose one of three settings:</p> <hr/> <p><b>Always Ask (Default):</b> Whenever a zeroing operation is required, the user will be prompted whether to perform the zeroing operation or not. Once this situation occurs, each subsequent measurement will cause such prompt until either the user allows the zeroing to be performed, or performs the zeroing operation manually from the GUI or changes the <b>Lambda Zeroing Mode</b> to either of the other two settings.</p> <hr/> <p><b>Automatically:</b> Whenever a zeroing operation is required, it is performed automatically right before the next measurement. This is the recommended setting for optimum lambda accuracy.</p> <hr/> <p><b>Manual Only:</b> Although a zeroing operation may be required, it is neither performed automatically, nor will there be a prompt, requiring a user response. Instead a notification will be shown in the measurement status box. The user will have to take care of the lambda zeroing.</p>
Use MPPM auto gain	<p>If checked, the measurement engine will configure any MPPM to use its auto gain feature, which gives improved dynamic range performance. Refer to the <i>N77xx User's Guide N7744-90B01</i>, the <i>N77xx Programming Guide N7744-90C01</i> and the Application Note "<i>Transient Optical Power Measurements with the N7744A and N7745A</i>"</p> <p>The default setting is on. Measurements of devices with high polarization dependence, especially &gt; 10dB, can be disrupted by the auto gain switching. Then it should be disabled.</p>

## Advanced Parameters

The Advanced Measurement and Post-Processing settings are located on the Advanced tab:

The screenshot shows three panels of settings:

- Advanced Measurement Settings:**
  - Store Measurement Raw Data in Files
  - Run Evaluation right after Data Acquisition
  - Maximum allowed reference loss (dB): 15.0
- Postprocessing Settings:**
  - Smoothing (nm): None
  - Generate TE/TM/Müller Data
  - Generate Diode Current Data
  - Generate Diode Responsivity Data
- Special Performance Settings:**
  - Timeout (min): 5.0
  - N7786B Quick Setup
  - Slow (Legacy) Mode
  - Quick SOP Search Mode
  - Fast Reference Mode

Buttons for Re-calculate, Light N7786B Init, Full N7786B Init, and TLS Init are also visible.

**NOTE**

The property **Reference Mode** has been removed in release 1.4, since a fully customizable referencing has been introduced. Click **Port/Ref. Manager** on the main tab to obtain and manage reference data for individual ports.

**Table 4** Application Fast IL/PDL Advanced Measurement Settings

Parameter	Description
Store Measurement Raw Data in Files:	If checked, measurement raw data is stored in saved files, in addition to evaluated data. This allows for later re-evaluation of the file with changed parameters, e.g. a different smoothing bandwidth. File size will be increased.
Maximum allowed reference loss (dB):	When performing reference measurements, the average power is compared with the TLS power set in the GUI. If this difference is greater than the maximum allowed reference loss, a warning is shown to the user, indicating e.g. bad optical connections.
Run evaluation right after data acquisition:	If checked (default), measurement data is evaluated immediately after data acquisition. Disable to increase measurement repetition rate. Measurement data can be evaluated later on if measurement raw data is stored in files. Note that this currently supports only wavelength-sweep related evaluations. It does not work with extinction ratio and dark current measurements active.

**Table 5 Application Fast IL/PDL Postprocessing Settings**

Parameter	Description
Smoothing:	Defines the width of the averaging window. Use smaller values for a higher wavelength resolution.
Generate TE/TM/Müller Data:	If checked, TE/TM traces and the first row of the corresponding Mueller matrix will be calculated during evaluation.
Generate Diode Current Data:	If checked, photodiode current traces will be calculated during evaluation. Option is only available if a power meter with electrical inputs or a B29xxA Precision Source / Measure Unit is used.
Generate Diode Responsivity Data:	If checked, photodiode responsivity traces will be calculated during evaluation. Option is only available if a power meter with electrical inputs or a B29xxA Precision Source / Measure Unit is used.
Generate Return Loss Data:	If checked, return loss traces will be calculated during evaluation. Option is only available if a return loss module is being used.

**Table 6 Application Fast IL/PDL Postprocessing Settings**

Parameter	Description
Re-calculate:	Press this button to re-evaluated the most recently obtained measurement data after changing any postprocessing settings, such as the smoothing parameter.
Full N7786B Init:	When using any direct instrument control to the N7786B instrument in between any two IL/PDL engine measurements, click this button once before starting the next IL/PDL measurement to make sure the instrument is initialized correctly.
Light N7786B Init:	This is similar to Full N7786B Init, but performs less initialization steps (faster). It's always safer to use Full N7786B Init, but under certain conditions, this may suffice.
TLS Init:	When using any direct instrument control to the laser instrument(s) in between any two IL/PDL engine measurements, click this button once before starting the next IL/PDL measurement to make sure the instruments are initialized correctly.

**Table 7 Application Special Performance Settings**

Parameter	Description
Timeout (min):	Determines the amount of time that the engine will wait for measurement data to be returned from the instruments.

**Table 8 Application Fast IL/PDL Special Performance Settings**

Parameter	Description
N7786B Quick Setup:	With PAS version 2.1, measurement initialization times of the N7786B have been optimized. Disable this to deactivate optimizations. Default is enabled. When disabling this, Slow (Legacy) Mode should be enabled instead.
Slow (Legacy) Mode:	With PAS version 2.1, general measurement initialization times have been optimized. Enable this to deactivate optimizations. Default is disabled. When enabling this, N7786B Quick Setup should be disabled instead.
Quick SOP Search Mode:	Reference measurement duration can be significantly decreased by activating the Quick SOP Search Mode. Note that this may affect the quality of the reference measurement, so a verification measurement right after taking a reference is highly recommended. Default is disabled.
Fast Reference Mode:	Reference measurement duration can be decreased by activating the Fast Reference Mode. Note that this may affect the quality of the reference measurement, so a verification measurement right after taking a reference is highly recommended. Default is disabled.

**NOTE**

When performing reference measurements with Fast Reference Mode disabled (default), the software may change the N7786B Quick Setup and the Slow (Legacy) Mode properties during the measurement. Once the reference measurement has been obtained, these settings are reverted to their initial states. Therefore, querying or modifying these properties through the automation interface while measurements are running is not recommended.

## Polarization Extinction Ratio Parameters

The Polarization Extinction Ratio settings are located on the Extinction Ratio tab.

Extinction Ratio Measurement Parameters						
	Wavelength Range (nm)		From Sweep	ER Samples	Power (dBm)	Powermeter Range (dBm)
SC-Band	1525.0	1550.0	<input checked="" type="checkbox"/>	3	10.0	0
CL-Band	1525.0	1550.0	<input checked="" type="checkbox"/>	3	10.0	0
			<input type="checkbox"/>			

**Table 9** Application Fast IL/PDL Extinction Ratio Parameters (per TLS)

Parameter	Description
Wavelength Range (nm):	Start and Stop Wavelength for (stepped) Polarization Extinction Ratio measurements. While <b>From Sweep</b> is checked, these settings cannot be changed.
From Sweep:	If checked, start and stop wavelength are taken from the main (sweep) measurement tab. Otherwise the values from the controls on this tab are used.
ER Samples:	Number of Extinction Ratio Samples to be obtained. When set to three, one PER measurement will be performed at the start wavelength, one at the stop wavelength and one at the center wavelength.
Power (dBm):	Defines the TLS power to be used for PER measurements.
Powermeter Range (dBm):	Defines the instruments' power range to be used for PER measurements. These controls will be disabled, while <b>Autorange</b> is set on the <b>Instrument Setup</b> tab. Note: When <b>High Capacitance Mode</b> is used or an B29xxA Precision Source / Measure Unit is part of the current configuration, Powermeter Range will only allow values of 0dBm or above.

## Performing a Reference Measurement

Before performing a single measurement or a series of measurements with similar settings, you need to take a reference. This is used to determine the optical power that will be applied to the input of the DUT during the TLS sweep, i.e., as a function of the TLS wavelength and the input polarization, as generated by the N7786B.

In the most simple case, the output from the N7786B is applied directly to one or more ports of the optical power meters. Select the desired measurement parameters first (see [Application Setup Parameters](#) on page 40). Some parameters are adjusted automatically, when starting the actual measurement, e.g. if the chosen wavelength range exceeds the available sweep range of the TLS, which also depends on the sweep speed, which is directly related to the step size setting.

The reference measurement must cover exactly the wavelength range that you are going to use when measuring the device under test and has to be taken with the same step size setting. Maximum accuracy is achieved if the TLS power during reference and DUT sweep is the same as well.

### NOTE

Performing a reference measurement is mandatory before starting the actual DUT measurement or measurement series.

---

### NOTE

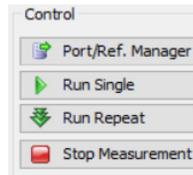
After performing a reference, the patchcord connecting the TLS with the N7786B must not be moved. When selecting other references from the drop-down list, make sure that the patchcord connecting TLS and N7786B has not been moved since taking that reference.

---

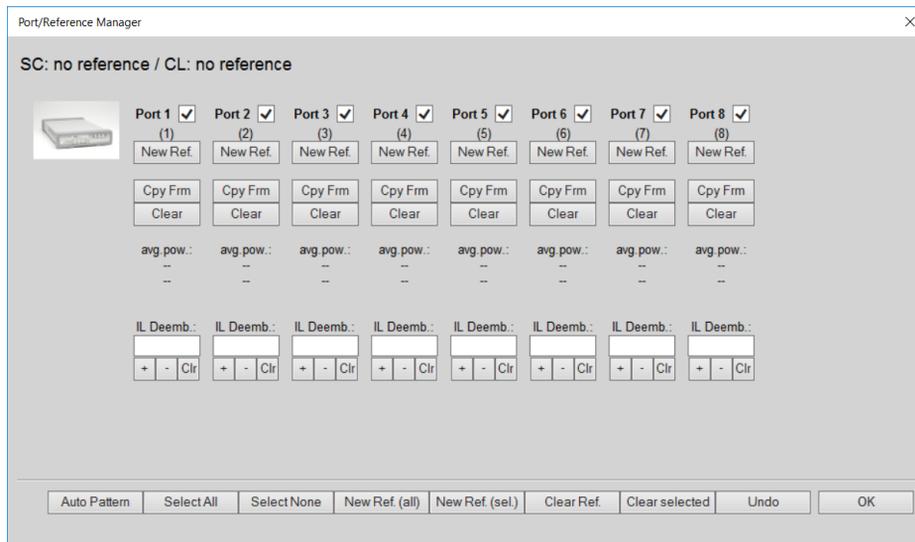
**NOTE**

When performing a reference measurement after clearing the previously loaded/measured reference or in case the new reference will be replacing all ports of the previous one, the software starts an optimization process for the N7786B polarization pattern. This optimization leads to a different pattern each time. Therefore, it's recommended to run a regular measurement right after obtaining that reference as a verification, which is expected to show reasonably flat IL and PDL traces. Such a verification measurement is especially recommended when using Quick SOP Search Mode or Fast Reference Mode (see Advanced Parameters section). When adding reference data for additional ports to an existing reference, no such optimization is performed. So usually, in these cases it is sufficient to verify the average power for each port as displayed in the Port./Reference Manager (refer the following figure).

To perform reference operations such as obtaining new reference traces for individual ports, copy reference data from one port to another or clear reference data from certain ports, click Port/Ref. Manager to open the Port/Reference Manager.



This dialog also displays helpful information about the reference data as described below. When you are finished checking or modifying the reference data, click the OK button on the bottom right to close the Port/Reference Manager.



The top row shows information about the TLS parameters (range, step size, TLS power). In a Multi-TLS application, the individual sets of parameters will be separated by slashes.

Below there will be a list of all ports of all power / current meter instruments configured for the application. To the left there is the instrument model code and serial number. To the right are all ports of the powermeter with their port numbers and check-boxes (described below). Beneath the physical port number of each power / current meter port there is a number in brackets. This is the virtual port number of each port, i.e., as if there was just one large array of power / current meter ports, no matter which instrument they are physically located in. This is the number shown as port number for measurements in the browser tree.

You can perform a reference measurement on a certain port by clicking the corresponding New Ref. button. You can do this for any number of ports, one after another.

Each channel that has been processed in this way will display the average power of the reference sweep on that port. In case of a Multi-TLS application, each line will correspond to one TLS, using the same order as the sweep parameter description at the top of the Port/Reference Manager.

**NOTE**

The wavelength sweep settings must be identical for all ports. If you change the sweep settings after obtaining reference data on some ports, the next reference sweep will clear the existing reference data in memory first. The engine will show a warning before clearing the data and allows to abort the operation instead.

---

**NOTE**

When there is no reference data available (e.g. the engine has just been installed, its wavelength sweep settings have changed or its reference has been cleared), the N7786B instrument settings will be optimized with respect to the input state of polarization. As mentioned before, the fiber(s) connecting the TLS and the N7786B must not be moved after taking a reference. So if the fiber has been moved, or has been subject to significant environmental changes, the N7786B setting optimization may become inaccurate.

Taking a new reference on only some of the ports will not necessarily cause the N7786B to be optimized again. Optimization will be performed only if either the reference was cleared before (**Clear** button), the reference is obtained using **Auto Pattern** mode (described below) or using **New Ref. (all)**. Optimization will also be performed if the next reference sweep will replace all currently existing reference traces simultaneously.

Whenever reference traces for at least one port remain unchanged by the next reference sweep, no optimization will be performed. (The intention is that the references on all ports use the same polarization states.)

As an indication you can check the measurement status box during the reference operation. If it is optimizing the N7786B settings, it will show “Optimizing Input SOPs” right after “Starting Reference Measurement”.

---

**NOTE**

The average power serves as a verification for the reference validity. Depending upon the TLS power selected in the GUI and the insertion loss of the reference connection, you should compare this value to the expected average power level.

---

You can perform a single measurement sweep on multiple ports simultaneously by using the check-boxes above the individual ports. All checked ports will be included in the reference sweep, once you click the New Ref. (sel.) button.

To obtain a reference on all ports simultaneously, click the New ref. (all) button. In this case, it doesn't matter which ports have been checked or unchecked.

Use the Select All and Select None buttons to set or remove all check marks simultaneously.

You can obtain reference data on one or more ports, then copy those traces to be used as reference for other ports. To do so, obtain the actual reference sweep, then check all destination ports (make sure other ports are unchecked) and click the Cpy Frm button at the source port.

All destination ports will show the detail information of the source port afterwards.

## NOTE

Especially when using N7745A specials with electrical input ports or B29xxA Precision Source / Measure Units, the **Cpy Frm** operation is required (or the **Auto Pattern** operation that is described below). A reference measurement is performed on one (or more) optical power meter ports, then the reference data needs to be copied to the electrical ports. To minimize the amount of data shown in the measurement results, reference data can be cleared on the optical ports after they have been copied to the electrical ports.

You can clear data for individual ports by clicking the Clear button of the respective port. By checking several ports and clicking the Clear selected button, reference data for those ports is cleared. By clicking the Clear Ref. button, the complete reference data is cleared from memory. Also upon restarting the engine, no reference data will be loaded automatically then, until a new reference has been obtained.

Actually the engine will still load reference files, but they don't contain any reference data.

## NOTE

Ports with a not-a-number indicator as power level in the reference details (or no average power indication at all) will not be evaluated during subsequent measurements.

There is also an automatic referencing mode available. After clicking Auto Pattern the engine checks all ports of all connected power / current meters for non-negligible power levels. If only one port shows significant optical power, this port's reference trace will be applied to all other ports later as well (as if taking a New Ref. on that one port, then using Cpy Frm to apply it to all other ports as well). If multiple power meter instruments are being used and one port of each of those shows significant power (e.g. two splitter outputs attached to any one port of each of two power meters), the respective reference traces will be applied to all ports of the respective power meter. If more than one port per instrument shows significant optical power, the sequence of reference traces found will be applied repeatedly for all ports of that power / current meter. For example one could connect the two output ports of a 50%-splitter to port 5 and 6 of a power meter and take a reference. The reference trace from port 5 will be applied to ports 1, 3 5 and 7, the reference trace from port 6 will be applied to ports 2, 4, 6 and 8.

You should have a look at the average power values of the power /current meter ports after performing an Auto Pattern reference to verify that the references have been applied to the respective ports in the expected way.

## NOTE

Please keep in mind that the **Auto Pattern** mode maps any reference traces found to all power / current meter ports in a repeating pattern.

For instance, if you connect two splitter outputs to port 1 and port 3 of a single power meter, take a reference in this mode and perform a regular measurement afterwards, the result will not apply the reference data obtained at port 3 to port 3 during a subsequent measurement.

Instead it applies the reference trace from port 1 to port 3 (as well as to port 5 and port 7). For doing this kind of measurement, you should use the custom referencing features described above, to make sure that each port is actually using it's own reference data.

---

You can load, save and delete reference files from the main GUI. See [Managing References](#) on page 58 for details.

**NOTE**

The check boxes are used for configuring reference sweeps only. If you perform DUT measurements subsequently, the check boxes will be ignored.

To exclude ports from evaluation, clear their respective reference data first.

---

**NOTE**

Since the reference data is obtained and required by the engine server, the files will be stored in a folder on the server PC, so no different folder may be selected.

This folder should be:

C:\ProgramData\Keysight\Photonic Application Suite\IL PDL Application\References\

When uninstalling the PAS main package, you will be prompted whether to uninstall or to keep application data for this engine, such as reference files.

---

Usually the last reference operation can be reverted by clicking the Undo button. Clicking this button will retrieve the reference state before the last reference sweep, clear, copy or load operation.

**NOTE**

When you start working with a certain setup and referencing mode, it is useful to perform a verification measurement after taking the first reference. Do this by either leaving the reference patchcord (or the splitter output pigtailed) connected to the ports used during the reference measurement or by connecting them to other ports that are supposed then to have valid references. Click **Run Single** and check the measurement results.

---

**NOTE**

When working with IL de-embedding, it is not necessary to modify the IL de-embedding file settings in Port/Reference Manager. Reference measurements will always ignore any IL de-embedding settings. See [IL De-embedding](#) on page 62 for further details.

## Managing References

If you run the IL/PDL engine for the first time and perform a reference measurement, the references will be saved automatically, using a default filename (one file per TLS).

You can save a reference by clicking the Save Reference button and load a reference by clicking the Load Reference button. Clicking these buttons will bring up corresponding file selection dialogs, showing details on the reference files.

**NOTE**

References are only useful when used for the TLS they have been obtained with. Make sure that you click the Load and Save button in the row corresponding to the correct TLS.

**NOTE**

An asterisk (\*) after the wavelength range indicates that the reference measurement has been performed with extended sweep range setting (see [Instrument Setup Parameters](#) on page 43) and requires subsequent measurements to be performed the same way and vice versa.

Filename	Date	Ch.	WLStart	WLStop	Rate	Power	LLog
130nmRange_SCL.pbin	2010/07/16 16:27:58	16	1500.0	1630.0	12.00	-1.0	No
130nmRange_SCL_1ch.pbin	2010/07/20 13:07:21	1	1500.0	1630.0	12.00	-1.0	No
130nmRange_SCL_2ch.pbin	2010/07/20 11:56:57	2	1500.0	1630.0	12.00	-1.0	No
20nmRange_B_CL.pbin	2010/07/07 10:32:04	1	1565.0	1585.0	6.00	-10.0	No
20nmRange_B_SCL.pbin	2010/07/07 10:30:48	1	1537.5	1557.5	6.00	-10.0	No
20nmRange_CL.pbin	2010/07/07 10:32:04	1	1565.0	1585.0	6.00	-10.0	No
20nmRange_SCL.pbin	2010/07/07 10:30:48	1	1537.5	1557.5	6.00	-10.0	No
20nmRange_SCL_PCanPort3.pbin	2010/07/15 13:16:59	1	1537.5	1557.5	6.00	0.0	No
20nmRange_SCL_singlecouplerport.p...	2010/07/15 15:15:39	1	1537.5	1557.5	6.00	0.0	No
CL_Ref3.pbin	2010/07/07 14:15:22	1	1565.0	1585.0	6.00	-10.0	No
CL_Ref4.pbin	2010/07/07 14:18:02	1	1565.0	1585.0	6.00	-10.0	No
ILPDLRef_TL50.pbin	2010/08/06 14:11:28	1	1537.5	1557.5	6.00	0.0	No
ILPDLRef_TL51.pbin	2010/08/06 14:12:20	1	1308.5	1328.5	6.00	0.0	No
O_806.pbin	2010/08/02 18:01:05	16	1308.5	1328.5	6.00	0.0	No
Ref1.pbin	2010/07/07 08:46:45	1	1537.5	1557.5	6.00	0.0	Yes
Ref2.pbin	2010/07/07 08:49:12	1	1537.5	1557.5	6.00	0.0	Yes

130nmRange\_SCL.pbin

Load Cancel

By clicking the Manage References button you can delete references. You can select multiple references at the same time by holding the shift or ctrl key while selecting the files.

### Performing a Measurement

After successfully completing a reference measurement or loading a reference file from disk, connect the output port of the N7786B to the input port of the DUT (usually using the reference patchcord or another measurement patchcord, to avoid excessive reconnection at the instrument). Then connect any output ports of the DUT to the power / current meter ports. Click Run Single or Run Repeat.

During a measurement you can abort the current measurement sweep and stop the repeat operation by clicking Stop Measurement. The measurement will be stopped as soon as possible, but usually not immediately, because some hardware operations have to be completed first.

### NOTE

If more than one TLS is being used, Stop measurement will abort the measurement for the current TLS sweep only. You may have to click Stop Measurement again for the remaining TLS.

**NOTE**

Measurement data will be returned for all ports that have reference data assigned (see [Performing a Reference Measurement](#) on page 51). Measurement data of other ports will be discarded.

---

## Details on PER and Dark Current Measurements

Dark Current and Polarization Extinction Ratio measurements are done prior to the actual sweep measurement if set to be included (see [Application Setup Parameters](#) on page 40 or [Reference: Interface "IEngine"](#) on page 104 for details on how to include those measurements).

Dark current measurements will not add graphs, since they return single scalar values for each measurement port. Dark current results are listed in an additional measurement info pane.

Polarization Extinction Ratio measurements will add a graph named "Extinction Ratio" (with a TLS ID suffix). In addition they will add an additional measurement info pane as described further below.

**NOTE**

Although Dark Current and Polarization Extinction Ratio measurements do not use reference data, they only return measurement results for those channels that have reference data assigned (see [Performing a Reference Measurement](#) on page 51). Hence, the results will match the port configuration of the sweep measurement.

---

**NOTE**

Dark Current and Polarization Extinction Ratio measurements are independent of any IL-de-embedding setting.

---

Polarization Extinction Ratio measurements performed with the IL/PDL engine are mostly intended to verify a certain minimum polarization extinction ratio of integrated receiver devices, typically around 20dB.

It is not intended for measuring very low PER/PDL values, since N7786B instrument PDL and measurement setup PDL are not taken into account. Typical PDL of the N7786B instrument is 0.25dB. Parasitic PDL of the measurement setup depends on the components used in the setup.

Fabry-Perot resonances (can occur when using multiple straight connectors) will have a similar impact on the results and should be avoided.

Polarization Extinction ratio measurements are done by creating a pseudo-continuous SOP pattern of a fixed length / duration, then evaluating maximum and minimum of the obtained power trace.

The higher the PER/PDL of the DUT, the higher the loss of the DUT, the lower the (effective) responsivity of the DUT and the closer the SOP pattern approaches the actual transmission minimum, noise and disturbances (intrinsic and induced) will have a stronger impact on the measurement result. To judge the impact of noise on the measurement, the measurement results also contain an uncertainty estimation, shown in an additional measurement info pane at the bottom of the GUI. The measurement data taken during the offset measurement, before the actual PER measurements, is evaluated for double standard deviation (assuming gaussian noise). The uncertainty value given in the measurement results is the DUT PER level for which the present noise would result in an uncertainty of +/-1dB.

#### NOTE

If the noise / disturbance cannot be assumed gaussian, the estimation will be incorrect. Therefore it is very important to avoid induced distortions, especially in case of small signals (high loss / low effective responsivity).

#### NOTE

Due to the logarithmic scale of the PER and the absolute level of noise and disturbances, measurements of PER values significantly above the value given from the uncertainty analysis may differ greatly between subsequent measurements.

#### NOTE

In case of very high actual DUT PER, due to the finite length of the SOP pattern used, the measured PER value may not be as high as the actual PER value. Since the SOP pattern depends on the input SOP of the N7786B instrument, the results may differ if the fibers have been moved or the temperature has changed, even if noise / disturbances are very low.

**NOTE**

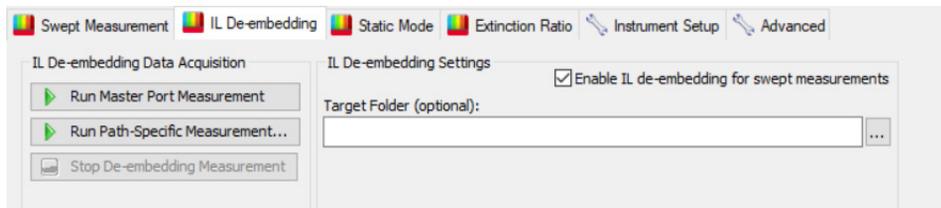
In case of high PER / high loss DUTs, better PER results may be achieved by using the High Capacitance Mode (see [Instrument Setup Parameters](#) on page 43), even if the DUT does not have a significant capacitance in parallel to the bias pins or is not a receiver at all.

## IL De-embedding

Using the IL De-embedding functionality of the IL/PDL engine, polarization-averaged insertion loss of components in the measurement setup, such as optical switches or splitters, as well as differences in the absolute responsivity of individual power meter ports can be characterized once and corrected for in subsequent IL/PDL sweep measurements. Insertion loss characterization data of individual components, such as wafer probes, can also be applied without prior characterization measurements, when available in OMR or CSV file format. The de-embedding data are generally used to remove IL differences in the measurement paths used for the reference measurement and DUT measurements. Multiple data files can be concatenated for the measurement to represent paths with multiple individually measured components.

**Characterizing Components / Obtaining IL De-embedding Data**

For obtaining component characterization data, there is a corresponding tab, the **IL De-embedding** tab, in the main engine GUI.



When obtaining IL de-embedding data, a master port measurement must be performed as a first step, then any number of specific path measurements may be performed.

Use the **Enable IL de-embedding for swept measurements** check box to enable/disable application of IL De-embedding data without having to clear all port-specific de-embedding file list controls in the Port/Reference Manager.

### Performing Master Port Measurements

A master port measurement is required as a reference for subsequent specific-path characterization measurements. To perform a master port measurement, connect the output of the N7786B Polarization Synthesizer directly to one of the power meter ports in the setup and click the **Run Master Port Measurement** button. (If future IL/PDL reference measurements will be made over a path including other components, like a switch, then this same path can be used for the master port measurement too, but this can make the concatenation of multiple elements in a path more complicated.)

#### NOTE

An optical signal should only be applied to one port, which will then be identified automatically.

---

#### NOTE

IL de-embedding measurements always cover the full range of all connected TLS(s) and automatically choose appropriate sweep rate and TLS power settings, so there's no need to adjust any settings on the Swept Measurement tab of the main GUI. Please be aware that settings on the Swept Measurement tab, as well as on the Advanced tab might change during the measurement, but will be reverted once the measurement is complete or aborted.

---

#### NOTE

Any master port measurement remains available during the current session. A new one needs to be obtained after restarting the IL/PDL engine, in case further setup-component characterization measurements are to be performed.

---

### Performing Specific Path Measurements

For specific path measurements, connect the output of the N7786B Polarization Synthesizer to the input of the instrument, component or path to be characterized, and the output of said device to the power meter port used for the master port measurement. Then click the **Run Path-Specific Measurement** button. You will be prompted to specify a name for the file

to be saved. The default folder for the file save dialog will be the one defined by the **Target Folder** control, unless the control is empty or describes a non-existent folder.

**NOTE**

Target file names must not contain any semicolon and must not start with plus (+) or minus (-) characters, as these characters are used for operation control, as described in [De-embedding Data for Setup Components](#) on page 64.

---

**NOTE**

Although not strictly necessary, using the same power meter port as for the master port measurement provides measurement data characterizing the path difference alone, without a contribution due to differences between power meter ports. Measurements for correcting differences in absolute responsivities of different ports (details described in [De-embedding Data for Cross-Port Responsivity](#) on page 66) can be made and applied separately for flexibility in configuring and concatenating IL de-embedding data for the optical paths used later.

---

**NOTE**

Since using the same power meter port during IL de-embedding master port and IL de-embedding specific path measurements provides de-embedding data independent of the port used, any subsequent IL/PDL reference measurements may be performed on any port as long as de-embedding data for all ports is considered correctly (see [Applying IL De-embedding Data](#) on page 64 for details). When creating additional files for correction of any differences between power meter port responsivities, the IL de-embedding master port should best be the one that is used for subsequent IL/PDL reference measurements (see [De-embedding Data for Cross-Port Responsivity](#) on page 66 for details), to simplify the required concatenations.

---

## Applying IL De-embedding Data

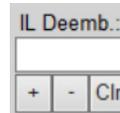
### De-embedding Data for Setup Components

Once IL de-embedding data for setup components is available, the Port/Reference Manager is used to configure which de-embedding files are to be applied to each power meter port's data.

## NOTE

Based on the state of the **Enable IL de-embedding for swept measurements** check box on the IL De-embedding tab of the main GUI, the controls described in this section might be disabled to indicate that no IL De-embedding will be performed unless the said check box state is toggled before any further measurement/recalculation.

For each power meter port, there is a configuration section for IL de-embedding settings.



The text control may be edited directly, using copy/paste or using the buttons below the text control.

In general, the text control defines a list of IL de-embedding files to be applied to the corresponding power meter port.

Filenames may consist of full path, file name and file extension, but for the sake of compact visualization and efficient editing, path and extension may be omitted.

File names without full path are expected within the folder defined by the **Target Folder** control on the IL De-embedding tab of the main GUI. It is also allowed to use sub-folders (plus filenames) here, which are then concatenated with the Target Folder settings.

For file names without extension, the IL/PDL engine will check whether a file of the respective name exists with either OMR or CSV file extension. If both exist, the OMR file will be used.

Each file in the list must be preceded by either a plus (+) or a minus (-) character, indicating the de-embedding operation to be performed. The corresponding operation is defined as follows:

Plus (+)	The device characterized by this file is present in the current path to this power meter port.
Minus (-)	The device characterized by this file was present in the path to the power meter port used for obtaining the currently active IL/PDL reference measurement used for this port.

As explained in [Performing a Reference Measurement](#) on page 51, reference measurement data is stored without applying any IL de-embedding data that might be configured in the Port/Reference Manager. Therefore, any devices present in the optical path during a reference measurement, must be corrected for in subsequent swept measurements by using the corresponding de-embedding files, preceded by a minus character.

Clicking the buttons with the plus or minus sign opens a file selection dialog and, unless canceled, adds the selected file using the corresponding prefix and a trailing semicolon to the corresponding port's de-embedding file list. Clicking the **Clr** button clears the corresponding port's de-embedding file list.

## NOTE

A semicolon is expected between any two file names, so filenames must not contain semicolon characters. A semicolon at the end of the file list is allowed, but not necessary. Multiple, adjacent semicolons are treated as one.

### De-embedding Data for Cross-Port Responsivity

While wavelength-dependent responsivities of the power meter ports are stored as part of the instrument calibration data and automatically corrected for during swept IL/PDL measurements, IL de-embedding may be used to compensate for responsivity differences between individual ports.

To do so, obtain an IL de-embedding master port measurement, as described above, then remove the fiber connector from the master port, connect it to another power meter port and run a specific path measurement. Repeat this for all power meter ports.

In the Port/Reference Manager, for each active port in your setup, add the corresponding cross-port IL de-embedding file to the IL de-embedding text control of the corresponding port.

**NOTE**

Keep in mind that for appropriate cross-port de-embedding, IL/PDL reference measurements must be obtained on the same port where the IL de-embedding master port measurement was obtained.

---

**NOTE**

If the Specific Path measurements above were made using power meter ports other than the one used for the master port measurement, then the Cross-Port Responsivity contribution is already included in those data and should not be added again in the IL de-embedding settings. But then that port is bound to that path so reconfiguration of the setup is more complicated and less flexible.

---

**IL De-embedding Settings during Recalculation**

When running an IL/PDL measurement with IL de-embedding settings active, a single, combined insertion loss trace is computed for each port and added to measurement raw data, unless raw data storage has been disabled.

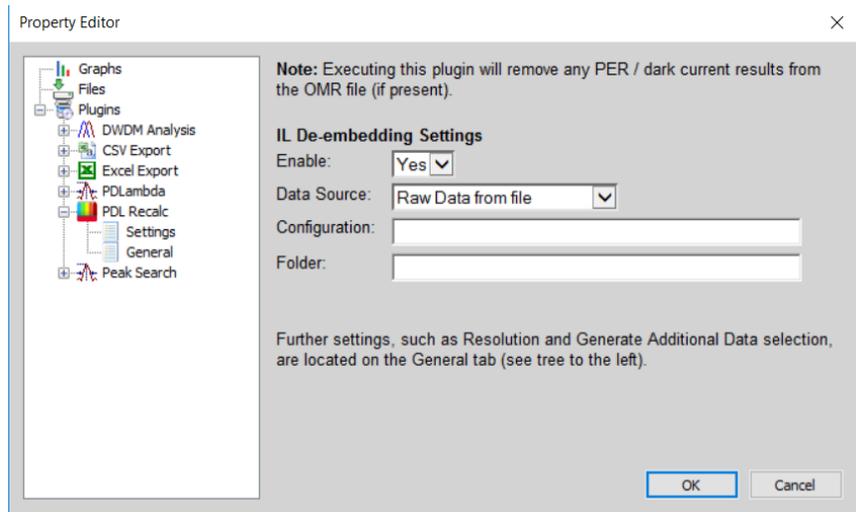
IL/PDL measurement OMR files, containing raw data, may be recalculated later, either from the File Viewer file context menu or using the OMR File Handler COM interface.

When using IL de-embedding files for correction of setup components, especially with many files involved, it is not too unlikely that there has been some error in the initial de-embedding configuration.

It is highly recommended to perform some verification measurements after the initial IL de-embedding configuration, to ensure the measurement results match expectations.

In case it still turns out, later, that there was something wrong with the IL de-embedding configuration settings, files may be recalculated, as described above, while modifying the IL de-embedding settings used.

The IL/PDL recalculation plug-in has four properties for controlling this.



- **Enable**

Use this toggle option to enable or disable IL De-embedding operations. If disabled, no IL De-embedding will be performed. If enabled, the **Data Source** control defines the actual operation to be performed.

- **Data Source**

Choose **Raw Data From File** to use exactly the IL De-embedding data that has been used during the operation that created the measurement raw data (usually the measurement itself or the most recent recalculation). Choose **Configuration String below** to ignore any IL De-embedding data stored with the measurement raw data and generate new IL De-embedding data based on the **Configuration** string defined in the plugin GUI. Please be aware that the latter operation will replace the initial IL De-embedding raw data, so be careful when saving the OMR file after applying this plugin.

- **Configuration**

This setting is used to override the IL de-embedding file list settings per port.

The format to be used here is a long string, with a sequence of port name / file list pairs. The port name should be three digits, encapsulated in two colons before and after. The file list is exactly the same string, as in the corresponding port's IL de-embedding control in Port/Reference Manager.

An example for a four port setup could look like this:

```
::001::-file0;+file1::002::-file0;+file2::003::-file0;+file3::004::-file0;+file4
```

in case the IL/PDL reference has been obtained on an optical path characterized by file0, which is then, for the actual IL/PDL measurement, replaced by four specific optical paths (described by file1 to file 4). It might also look like this:

```
::001:::002::-file0;+file2::003::-file0;+file3::004::-file0;+file4
```

in case the IL/PDL reference has been obtained on port 1, with exactly the same components in the optical path, as later on, during an actual IL/PDL measurement (note that a sequence of four colons is actually the trailing pair of the former port and the leading pair of the latter port).

- **Folder**

This setting is identical to the **Target Folder** setting on the IL De-embedding GUI tab. It may be used in case the IL de-embedding files are available, but they have been placed in a different location than during the initial measurement.

In the **Settings** tab of the Measurement Info Pane, the IL de-embedding configuration string used during initial measurement and actual measurement evaluation / recalculation can be checked / extracted.

### File Formats

While the IL de-embedding measurement of the IL/PDL engine can be used to characterize setup components, it is also possible, to use existing OMR or CSV files for IL de-embedding operation.

In case of OMR files, it is important that there is an “Average IL (TLSx)” trace present, as is the case for IL/PDL engine measurements, or an “IL” trace, as is the case for IL engine or FSIL engine measurements. When accessing OMR data through the OMR File Handler COM interface, the corresponding trace name is RXTXAvgIL or TLSx\_RXTXAvgIL.

The IL de-embedding data wavelength range must exceed or at least match the wavelength range of the current measurement.

## NOTE

In case of multi-TLS IL/PDL measurement files, there is a single IL trace for each TLS involved and the appropriate one is used automatically.

When using CSV files, highest evaluation speed is achieved if both the PC running the IL/PDL evaluation and the CSV file use the same decimal and list separators characters. The IL/PDL evaluation should be able to cope with mismatching settings between operating system locale and CSV file used, but might require some additional time for data conversion.

CSV files for IL de-embedding must consist of exactly two data columns, namely the wavelength in meters and the insertion loss (IL) in dB.

**NOTE**

When viewing OMR measurement files in File Viewer (or in measurement engine GUIs), the y-axis is mirrored, i.e., negative numbers above the origin and positive numbers below. This is to visualize more intuitively that larger insertion loss means weaker transmission / less received power. So, the IL data in OMR files is in fact loss data, not transmission data and, correspondingly, CSV files used for IL de-embedding must contain loss data (positive values for attenuation, negative values for amplification).

---

**NOTE**

CSV files must be provided in plain ASCII format, i.e., without, for example, a UTF-8 header. When saving CSV file from Excel, please be aware that Excel may provide multiple CSV file formats (including CSV UTF8, depending on the Excel version). Be sure to use “CSV (Comma delimited)”, if available. Please note that the UTF-8 header (as well as other encoding headers) are usually not shown, when viewing the file. In case a CSV file causes trouble that might be caused by an invalid encoding, open Notepad.exe, load the offending file and save it using the “ANSII” option.

---

**IL De-embedding Data Cache**

For maximum evaluation speed, the IL/PDL engine stores internal files, containing IL de-embedding data from one or more source files, after merging and interpolation to target wavelength grid. Whenever the same list of source files is used later, for the same wavelength settings, the stored data is used instead of repeating merge and interpolation operations. The location on the disk drive that contains these internal files is called the IL De-embedding Data Cache.

Files in the IL De-embedding Data Cache are not deleted automatically. In case there is any IL De-embedding data at all stored in the cache, during engine activation, there will be a message in the status log, indicating the amount of used disk drive space.

Once the number of file exceeds 1000 or the combined size of all the files in the IL De-embedding Data Cache exceeds 1GB, there will be a warning message, suggesting to manually clean up the IL De-embedding Data folder.

To perform this task, navigate to

C:\Users\All Users\Agilent\Photonic Application Suite\IL PDL Application\ILDeembeddingDataCache

and delete files there.

## NOTE

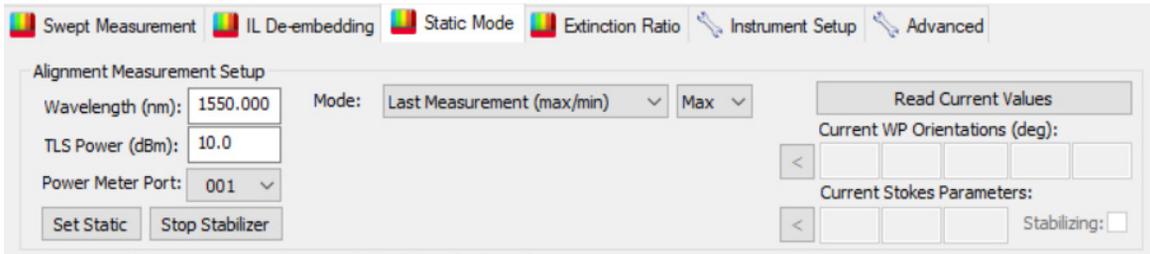
File names are automatically generated and cannot be used to identify contributing IL de-embedding files or wavelength range settings. In case you are dealing with many different and large IL de-embedding files, consider a recurring clean-up operation of just deleting all the files in the IL De-embedding Data Cache. The IL/PDL engine will then create new files, whenever new configurations are being used.

## Static Mode

While swept IL/PDL measurements are a quick and robust method for characterizing polarization-dependent and polarization-averaged transmission properties of passive optical devices and receiver sub-assemblies, sometimes specific custom measurements might also be required for certain devices.

To support this kind of scenario, the IL/PDL engine provides the Static Mode, which combines a variety of ways for setting a specific state of polarization at the DUT.

The Static Mode is configured on a separate tab, the **Static Mode** tab, in the main GUI.



Static Mode operation is configured largely independent of any other IL/PDL engine measurement mode, except for selecting the TLS output port (in case there is more than one), which must be done on the Swept Measurement tab of the GUI.

The Static Mode parameters and buttons are explained in the table below.

Parameter	Description
Wavelength (nm)	Defines the wavelength to be set once the <b>Set Static</b> button is pressed. For the <b>Last Measurement</b> and <b>OMR File min/max</b> modes, it also defines the wavelength at which the measurement data is to be evaluated for min/max transmission or PSPs.
TLS Power (dBm)	Defines the optical output power of the laser source.
Power meter port	Defines the power meter port to be evaluated in OMR measurement data when using <b>Last Measurement</b> and <b>OMR File</b> modes. The drop-down menu lists power meter serial number and power meter port number for reference. When using <b>OMR File</b> mode, make sure to only use files that have been obtained using the same power meter configuration that is currently active.
Set Static	Activates Static Mode using the current settings. Unless Mode is set to <b>Waveplate Orientations</b> , <b>Set Static</b> starts the polarization stabilizer operation of the N7786B, with a target State of Polarization (SOP) based on the Static Mode configuration. With Mode set to <b>Waveplate Orientations</b> , stabilizer operation is neither required nor supported.

Parameter	Description
Stop Stabilizer	Clicking this button will make sure that the stabilizer operation of the N7786B is stopped, in case it is active. If clicked even when the stabilizer was not active, it has no effect, so it may always be used to make sure the stabilizer is stopped. Stopping the stabilizer after it has found the desired target SOP might be desired, to avoid mild SOP variation due to stabilizer dithering. But the SOP can then drift or change if the polarization at the N7786B input changes or due to temperature variation. It is not necessary to stop the stabilizer manually, before starting a swept IL/PDL measurement.
Mode	Defines the mode of operation for Static Mode. The different behavior of these modes and the corresponding GUI changes are described below this table.
Read Current Values	Clicking this button reads current SOP and current waveplate orientations, as well as the current stabilizer state. Details are described below.

### Static Mode Modes

Consider the following aspects when using Static Mode in general:

- It is useful, although not mandatory, to use a PMF between TLS and N7786B and it is highly recommended to use an SMF between N7786B and DUT, except for PMF that is considered to be part of the DUT.

Consider the following aspects when using any Mode other than Waveplate Orientations:

- Once **Set Static** is clicked, the polarization optimization is started. The engine / GUI remains busy until the target SOP has been reached or an unexpected event has occurred, such as an instrument timeout or an error caused by a too low input power level. After the SOP has been found, the stabilizer operation continues, until manually stopped or until another kind of measurement is performed from within the IL/PDL engine.
- Stokes parameters are defined relative to the reference system at the polarimeter inside the N7786B, not relative to any DUT main axes or geometries.
- Polarization stabilization works based on SOP measurements within the N7786B. Therefore, polarization changes in front of the N7786B can be compensated for, but polarization changes between N7786B and DUT cannot. Accordingly, connecting fibers should be fixed or taped in place and should not be subject to movement/vibrations or temperature variation.

- When using **PDLSP** mode (as described below), assignment to PSP1 or PSP2 is arbitrary to some degree. In case of stable DUT and setup and when using the same IL/PDL reference, association should be persistent between measurements, but in case any of these requirements do not apply, the assignment may swap between measurements. Assignment will never toggle within a single measurement sweep, but PSP orientation may gradually transform over wavelength, in case there is strong birefringence in the measurement Setup or DUT (with main axes other than the PDL main axes).

Consider the following aspects when using Waveplate Orientations Mode:

- Using **Waveplate Orientations** Mode, there is no stabilizer operation at all. This is a pure feed-forward operation. For example, if the SOP input to the N7786B changes, the output SOP will also change.

The different available modes are described in the following sections.

#### **Last measurement (min/max) or (PDLSP)**

When Mode is set to **Last Measurement**, a swept IL/PDL measurement must first have been performed since starting the IL/PDL engine.

Once **Set Static** is clicked, the current measurement data, i.e., data from the most recent measurement sweep, is analyzed on the selected port and the resulting SOP is set as target SOP and polarization stabilizer operation of the N7786B is started.

In **min/max** mode, based on the min/max selector, the corresponding SOP is extracted, to provide minimum/maximum transmission at the selected wavelength.

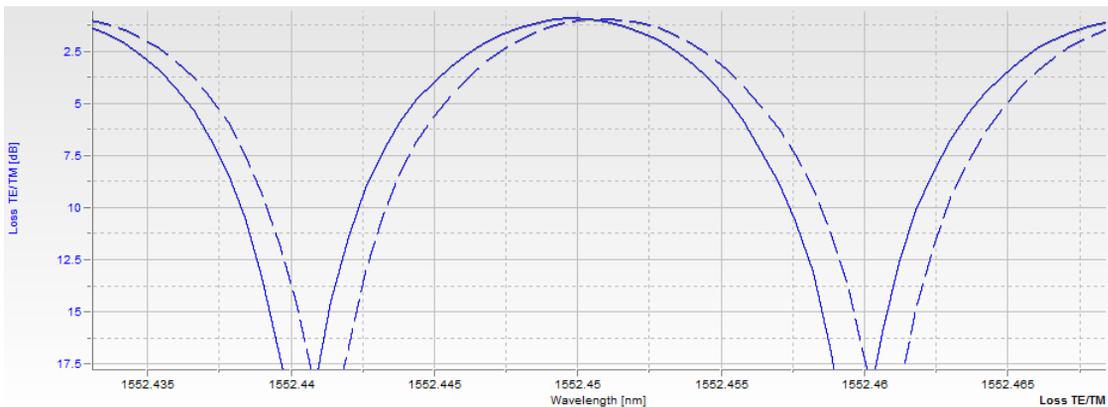
In **PDLSP** mode, the complete measurement range is analyzed in the same manner that leads to TE/TM traces in a regular IL/PDL measurement; an appropriate wavelength region of high DUT PDL is selected automatically, then the corresponding principal state SOPs are extracted and applied based on the PSP1/PSP2 selector.

### **NOTE**

**Measurement traces are named TE/TM for distinction. They do not necessarily refer to physical definition of TE/TM, but might be swapped instead.**

The difference between TE/TM and min/max traces is most prominent in certain filter devices, where there can be a wavelength-shifted filter response for two orthogonal polarizations. This filter shift is, obviously, discernible when sequentially sweeping the wavelength after aligning the input polarization to the DUT's principal axes. It is also obtainable from a

single-sweep IL/PDL measurement (TE/TM trace), as shown in the graph below. As the graph indicates, using PDLPSP mode will use SOPs matching the DUT's main axes and, thus, return maximum or minimum transmission, based on the chosen wavelength. Using min/max mode, on the other hand, would align to either of the main axes, when far from the filter's passband, but might align to some average SOP close to its passband center, where the response curves cross.



## NOTE

Last Measurement Mode requires Mueller data to be present in the measurement data. This requires Generate TE/TM/Mueller Data to be checked on the **Advanced settings** tab.

### OMR File (min/max) or (PDLPSP)

This Mode is basically identical to **Last measurement (min/max) or (PDLPSP)** on page 74, except that it does not extract the min/max/PSP information from the current measurement, but rather from a saved file.

When this Mode is selected, there will be additional controls for either typing or copy/pasting a filename (including full path and file extension) of an OMR file, or opening a file selection dialog to choose such an OMR file.

Mode:

Filename:

**NOTE**

Make sure to only use files that have been obtained using the same power meter configuration that is currently active.

**NOTE**

**OMR File Mode** requires Mueller data to be present in the measurement data. This requires **Generate TE/TM/Mueller Data** to be checked on the Advanced settings tab during measurement/recalculation.

**Stokes Parameters**

In **Stokes Parameters** Mode, the target SOP is provided using its normalized Stokes parameters.

Mode:	Stokes Parameters			
S1:	S2:	S3:		
1.00	0.00	0.00	Set orthogonal	

These parameters can be obtained, e.g., from polarimetric measurements using Polarization Navigator or AgN778xLib instrument automation or using the Get Current Values operation described in [Current Value Operations](#) on page 77.

Use the **Set Orthogonal** button to switch to the orthogonal SOP, for example, after having manually optimized for one of the DUT's main axes before.

**Waveplate Orientations**

In **Waveplate Orientations** Mode, the target orientations for the five equivalent waveplates of the N7786B instrument are provided in degrees.

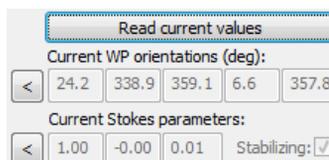
Mode:	Waveplate Orientations				
WP1 2 3 4 5 orientation (deg):					
0.0	0.0	0.0	0.0	0.0	0.0

**NOTE**

There is no polarization stabilization involved when using **Waveplate Orientations Mode**.

### Current Value Operations

To check the current SOP or waveplate settings, as well as the current polarization stabilizer state, click the **Read Current Values** button.



Current values may result either from a previous Set Static operation or an operation performed independent of the IL/PDL engine, for example, by using Polarization Navigator or AgN778xLib automation. After reading the current values, the buttons to the left of the respective waveplate or Stokes values can be used to make these values the new target values and switch the Mode to **Waveplate Orientations** or **Stokes Parameters**, respectively.

**NOTE**

Reading current values requires some reinitialization before the next IL/PDL measurement sweep, so some additional time is required. This does not matter if **Set Static** has already been performed, since the measurement sweep needs to be reinitialized then, anyway. However, it is not recommended, for example, to run **Read Current Values** in between every two IL/PDL sweep measurements, unless explicitly required.

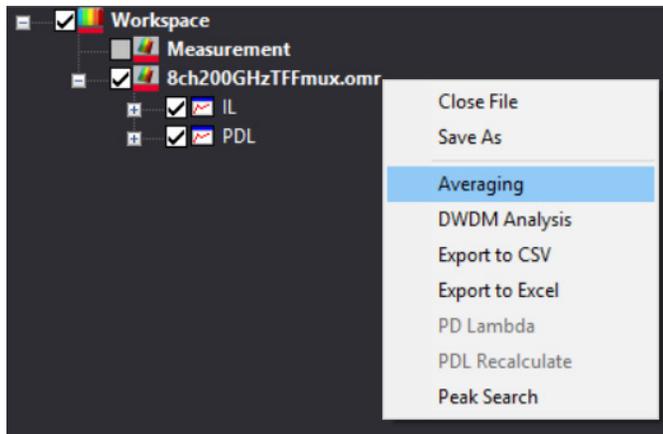
### Re-Evaluating Measurements

The current measurement can be re-evaluated with changed parameters, e.g. a different smoothing width or after changing the state of any of the Generate ... Data check marks on the Advanced tab.

Change the respective parameters and click the Re-calculate button on the Advanced tab to perform a re-evaluation.

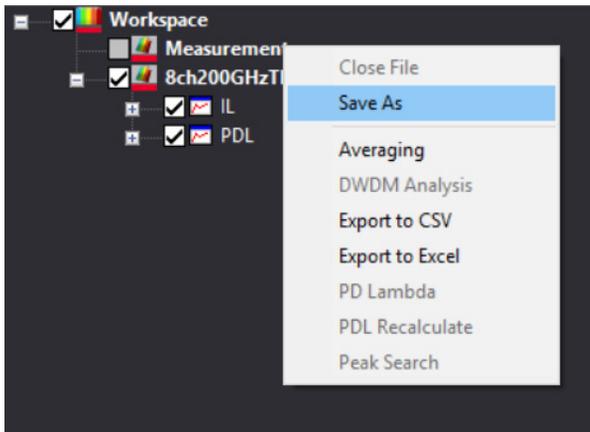
## Applying Plugins

You can apply evaluation plugins to the current measurement, e.g. averaging, peak-search or DWDM channel analysis functions. The plugins are listed in the measurement context menu in the browser tree. Which plugins are available depends on the license(s) that are installed on your system. For more information about the available feature packages and licenses, refer to [Keysight N7700 Photonic Application Suite - Brochure](#). Some plugins directly modify the traces of the measurement, some add more traces, and some add information to the table on the bottom of the user interface.



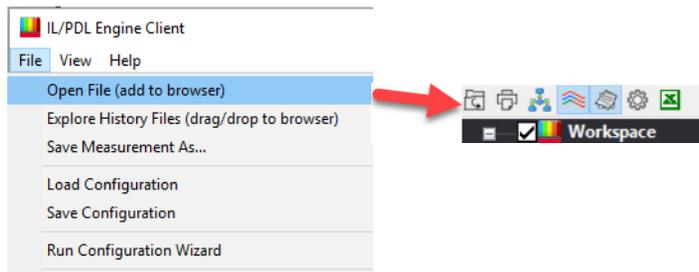
## Saving Measurements

You can save measurement data using the Save as option from the measurement context menu in the browser tree and providing a filename. Alternatively, select **File > Save Measurement As...** The measurements are saved as .omr files, which can be opened with the same N7700 engine, with the N7700 File Viewer or with customer programs using the N7700 OMR File Handler library.



### Loading Measurements

Saved files can be accessed through the Open File dialog from the File menu or by clicking the Open icon above the browser tree. Open files will be listed in the browser tree to the left of the user interface.



### Creating .agconfig Files from Measurement Files

You might require the .agconfig file for a particular measurement that you have performed in the past. In case you did not explicitly save the .agconfig file, you can extract it from a measurement (.omr) file by using ExtractAgconfigFromOmr, a PowerShell utility, available in the Examples folder (C:\Program Files\Keysight\Photonic Application Suite\Examples\DemoPowerShell). This utility takes an OMR file as an input and creates an agconfig file that reflects the settings with which the specified

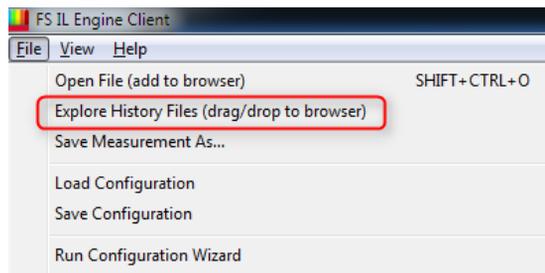
measurement had been performed. The extracted .agconfig file has the same name as the specified OMR file and is created in the same directory as the OMR file. For easy access and execution of this PowerShell utility, use the batch file ExtractAgconfigFromOmr.bat specifying the OMR file as a parameter (either specify the full path of the OMR file or place it in the current working directory).

This utility can also be used for troubleshooting purposes.

### Automatic History Saves

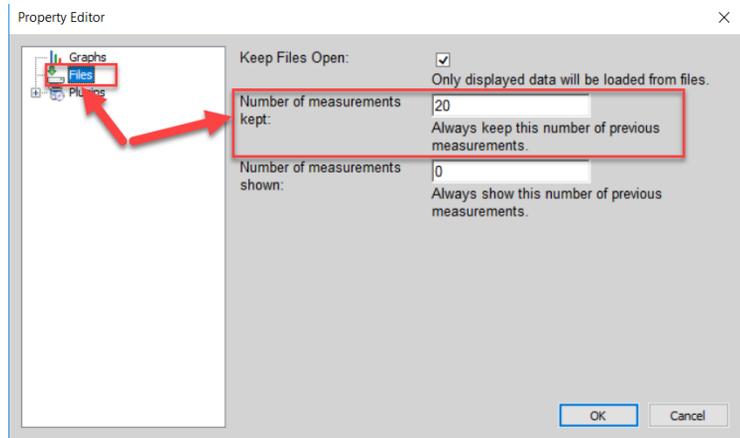
By default, the measurement engine will automatically save a number of recent measurements to a specific folder. This is a very helpful feature, since one often performs a number of measurements that are not intended for saving, then realizes, after a certain, subsequent measurement, that comparison with one of the previous measurements would indeed reveal certain insights.

Selecting Explore History Files from the File menu will open this engine's history file save folder. The files are named using the engine acronym followed by a time stamp. Use drag and drop to add any file(s) to the engine's browser tree or double-click any file(s) for display in the File Viewer application.



Especially when working with large amounts of measurement data (many ports, broad range, high resolution), you may want to reduce the number of automatically stored files or even turn off the automatic saving of history files. On the other hand you may want to increase the number for obtaining many measurement files without having to perform manual saves, or having to use measurement automation.

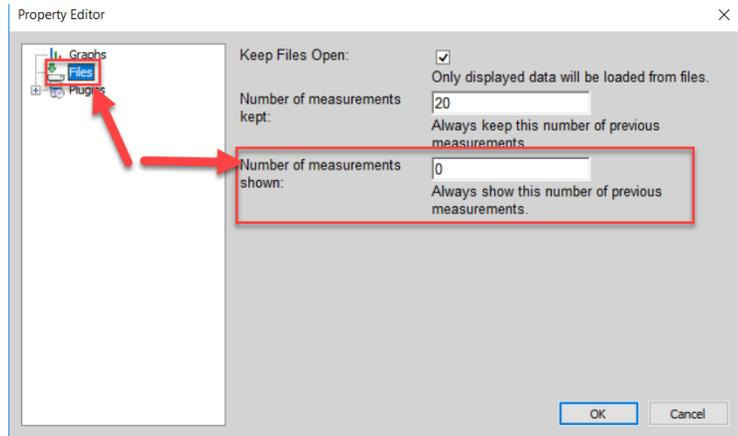
You can do so by modifying the corresponding value in the options:



Set the **Number of measurements kept** value to 0 to turn off automatic saves.

When tuning a device or a component, or monitoring stability, it might be desirable to automatically have a certain number of most recent measurements displayed simultaneously.

To do so, set the **Number of measurements shown** value in the options to the desired value or to 0 to disable this feature.

**NOTE**

**Number of measurements kept** must be larger than or equal to **Number of measurements shown**, for this feature to work as intended.

**NOTE**

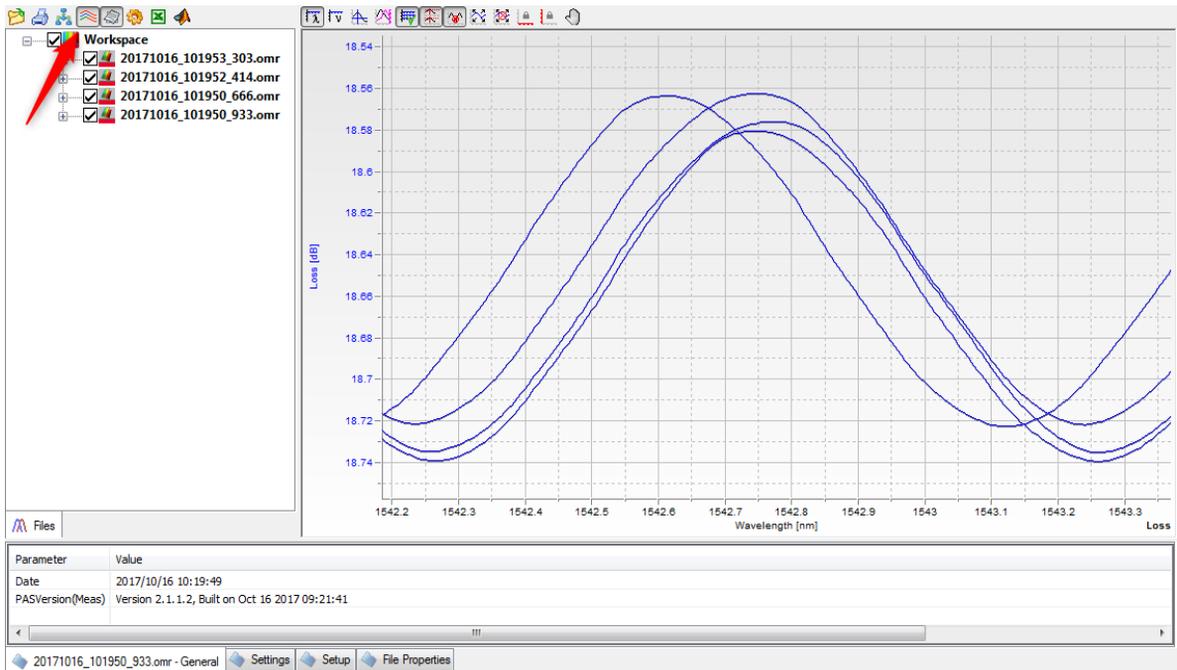
**Number of measurements kept** and **Number of measurements shown** are global settings used for all Photonic Application Suite engines.

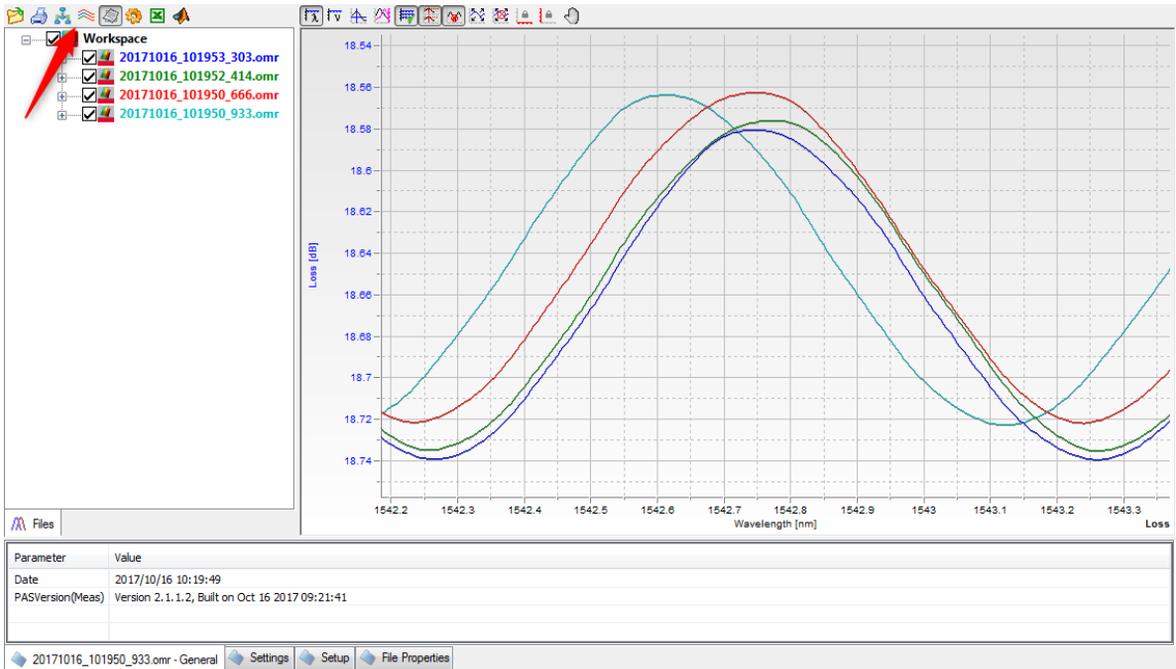
**NOTE**

The **Number of measurements shown** setting affects measurements performed after starting the engine, after changing this setting, or after closing all the open files. It does not open any older files, so after any of the mentioned operations, the number of automatically shown history files will grow with each new measurement until the total number reaches the **Number of measurements shown** value. For each new measurement after that, the oldest history file will be closed automatically.

By default, trace colors indicate different ports, but by toggling the **Color by Channel/File** icon, this can be changed to indicate different measurements instead. This is usually desired, when using automatic history file display with single-port devices, or devices where individual ports are clearly distinguishable.

See examples for both modes below.





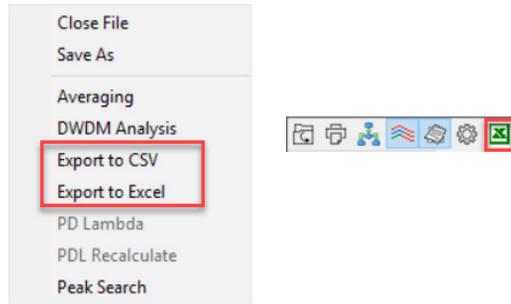
When either replacing the DUT, introducing a significant change to DUT tuning parameters, or running a specific path de-embedding measurement, corresponding measurement results may vary significantly from previous history files. To remove all the old measurements from the Workspace at once, click **Close All** from the context menu of the Workspace node in the tree on the left of the GUI.

## NOTE

History files are stored in an engine specific subfolder of  
 Windows 7 and above: C:\Users\

## Exporting Measurement Data

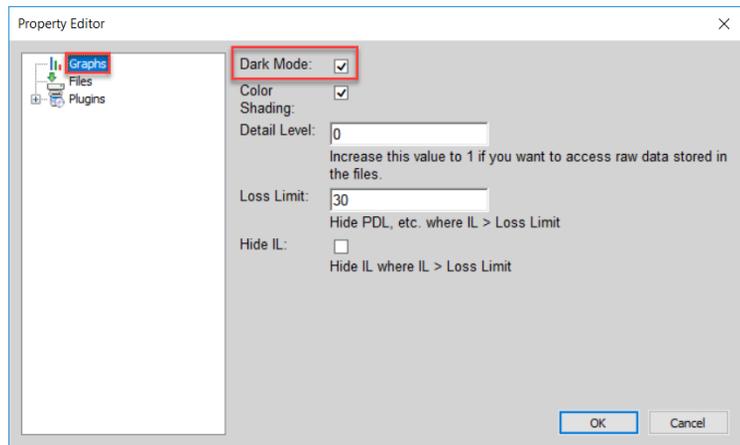
Measurement data can be exported to different applications or file types (e.g. Excel, CSV). You can export the current measurement by selecting the corresponding entry of the measurement context menu in the browser tree or the corresponding icon above the browser tree.



## Changing the Color Theme

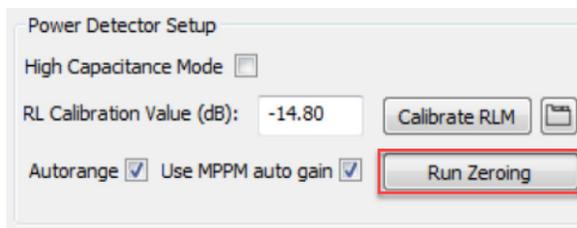
You can switch to a lighter color theme for the File Viewer within the user interface of IL/PDL Engine. Click the Settings icon and disable the Dark Mode check box in the Graphs section.





### Power/ Current Meter Zeroing

Use the Run Zeroing button to zero all ports of all connected power / current meter instruments, as well as an 81610A or 81613A return loss module, if used. You will be prompted to cover all detectors before proceeding. The TLS output is turned off automatically during zeroing. Optimum dynamic range and sensitivity is achieved, after power / current meters have been zeroed appropriately. The instruments should have reached stable operating temperature conditions for best results.



### TLS Lambda Zeroing

In order to ensure optimum wavelength sweeps through varying ambient conditions, a lambda zeroing function is available.

**NOTE**

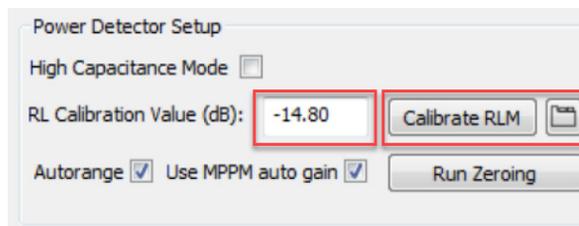
For maximum wavelength accuracy, a lambda zeroing operation should be performed, once the instrument reached stable operating temperature conditions.

## Return Loss Calibration

By default, the return loss module will be operated using the calibration parameters currently stored on the instrument (wavelength-independent), not yielding optimum accuracy across the whole wavelength sweep.

Therefore, it is recommended to perform calibration steps for the return loss module. This can be done using a calibrated or known reference reflector such as the 81610CC listed in the RL module user's guide.

Set the RL Calibration Value (dB) on the Instrument Setup tab to match your reference reflection artifact. Then use the Calibrate RLM button to start the return loss calibration procedure. The application will indicate when to connect the reference reflector to the return loss module output and when to connect or make a termination artifact.



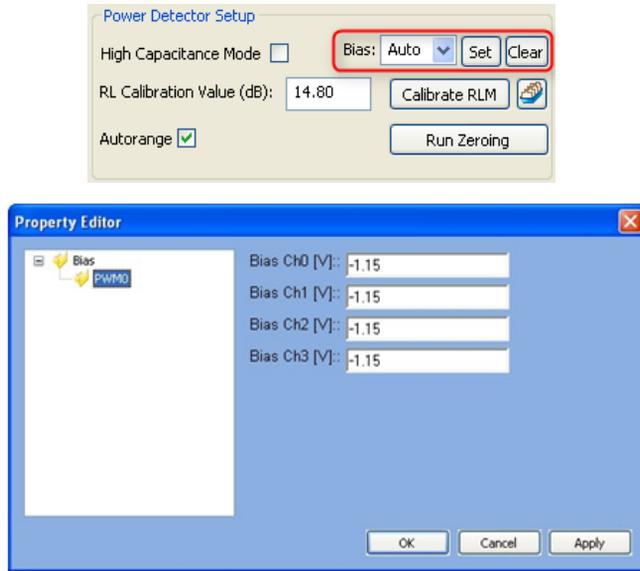
Calibration will be performed for all TLS currently checked in the Swept Measurement tab. The calibration range will cover the available sweep range of the laser sources, so it is unnecessary to repeat the calibration when changing the measurement parameters (wavelength start, wavelength stop or sweep rate).

The calibration data will be saved automatically and applied to all subsequent measurements until a new calibration is performed. The engine will show a message in the measurement status window, telling the user whether a calibration file has been used, or whether the wavelength independent calibration parameters stored on the instrument have been used.

By clicking the square button next to the Calibrate RLM button you can access calibration files for deletion.

## Bias Settings

If the setup comprises at least one current meter or power meter with electrical inputs for measuring photocurrents, the Instrument Setup tab will show buttons for setting and resetting the bias values.



Clicking the Set button will bring up a dialog displaying the current Bias settings. Click Cancel to exit the dialog without changing the settings. Click OK to apply changed bias values.

Clicking Clear will set the bias voltages to 0V for all electrical input ports.

The Bias drop-down list determines the bias mode as described in the following table.

<b>Auto (Default)</b>	Actual bias voltages are kept at 0V until the measurement is started. During the measurement they are set to the defined values automatically. After the measurement has finished, actual bias voltages are reverted to 0V.
<b>Off</b>	Actual bias voltages are set to 0V, regardless of the defined bias voltage settings.
<b>On</b>	Actual bias voltages are set to the defined bias voltage settings right away and kept throughout, as well as after the measurement.

### Configuration Handling

The current configuration, i.e. hardware setup as well as current parameters like wavelength range, can be saved and loaded from the File menu by selecting Save Configuration and Load Configuration. A saved configuration file (\*.agconfig) can also be started from the Windows Explorer and will launch the IL / PDL engine if it's not running. The configuration wizard can be launched from the File menu as well.

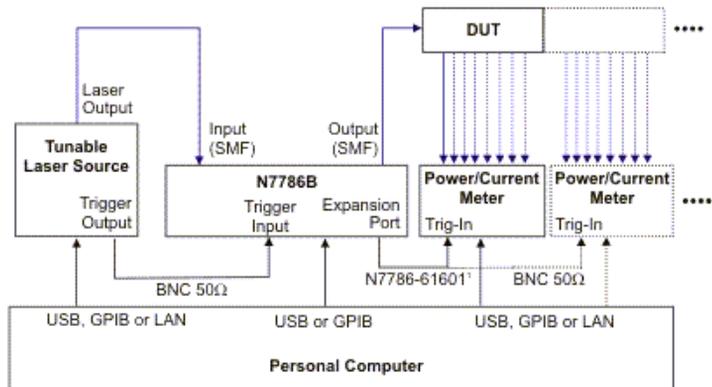
#### NOTE

If an agconfig file with identical hardware settings is loaded from within an activated engine, only the changed measurement parameters will be applied to the current engine. You can use this to quickly switch between different parameter sets.

If an agconfig file with different hardware settings is loaded instead, the current engine will be deactivated and the chosen agconfig file will be used to activate a new engine.

### Basic Measurement Setup

The general setup is illustrated in the following diagram:



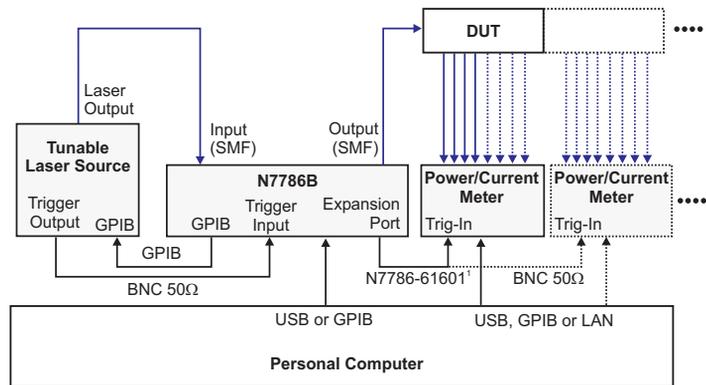
\*: N7786-61601 is color coded with red heat shrink tubing at both ends

## NOTE

Make sure that to perform the software installation steps described in [Software Installation](#) on page 28 before connecting the instruments.

- Connect the N7786B to the PC either using the GPIB or the USB interface.
- Connect any N7744A / N7745A / N7744C / N7745C / B29xxA that you want to use with this application to the PC using the USB, GPIB or LAN interface.
- Put any 81636B that you want to use in the TLS mainframe.

- Connect the TLS mainframe to the PC or to the N7786B (Remote-GPIB) using the GPIB or LAN interface. The Remote-GPIB Setup is shown in the following diagram:



<sup>1</sup>: N7786-61601 is color coded with red heat shrink tubing at both ends

## NOTE

The Configuration Wizard will ask whether to use Remote-GPIB or not. For further information refer to [Using the N7786B GPIB Interface \(Remote GPIB\)](#) on page 31.

- Connect the Trigger Output of the TLS with the Trigger Input of the N7786B (using a 50 Ohm BNC cable).
- Connect the N7786-61601 Trigger Cable to the Expansion port of the N7786B and to the Trigger Input of one of the power / current meters. The N7786-61601 Trigger Cable is marked by red heat-shrink tubing at both ends.
- When using N7744A / N7745A / N7744C / N7745C instruments, trigger signals at the trigger input will be routed internally to the trigger output, so daisy-chaining is possible. (using 50 Ohms BNC cables).
- When using B29xxA Precision Source / Measure Units or 81636B Power Sensors, coaxial tee connectors are required to route the trigger signal to all power / current meters. (see [Trigger Connections](#) on page 17)
- Connect the optical output of the TLS with the optical input of the N7786B using a patchcord with appropriate connectors (angled or straight).

- Connect the optical output of the N7786B with the optical input of the device under test (DUT) using a patchcord with appropriate connectors (angled / straight).
- Connect any optical outputs of the DUT to the optical inputs of the N7744A / N7745A / N7744C / N7745C / 81636B. Connect any electrical outputs of the DUT to the electrical inputs of the N7745A-E01/-E01 or B29xxA instruments.
- For the reference measurement, the fiber or fibers that will be applied to the DUT input later are connected to optical power meter ports directly first.

**NOTE**

The fiber connecting TLS and N7786B should be carefully positioned and secured for stability without movement or vibrations.

**NOTE**

Optimum performance is obtained when power / current meters (and return loss module, if used) have been zeroed appropriately (see [Power/Current Meter Zeroing](#) on page 86).

## Extended Measurement Setup

The IL/PDL engine may be used with up to three tunable laser sources, covering multiple wavelength bands. If more than one TLS is used, an optical switch module, like the 81595B, is required in one of the mainframes as well. In addition to supporting multiple N7744A/N7745A/N7744C/N7745C/81636B power meters and B29xxA Precision Source / Measure Units, an optical return loss module 81610A or 81613A can be used.

For details on such extended setups, refer to section [Fast Multichannel IL / PDL Setup](#) on page 15.

# 4 Automation

COM Components	/ 94
Creating a New Engine	/ 96
Connecting to an Existing Engine	/ 98
Performing a Measurement	/ 99
Accessing the Measurement Result	/ 100
Reference: Interface "IEngineMgr"	/ 102
Reference: Interface "IEngine"	/ 104
Reference: Interface "IEngineTLS"	/ 135
Alphabetical Automation Index	/ 144

## COM Components

Automation is implemented using a mechanism called “COM”. COM has been introduced by Microsoft on the Windows platform to allow a unified way to communicate between different software components. Today, almost every programming language such as C#, C++, LabView, Keysight Vee as well as MATLAB offers ways to use these so-called COM components. Once familiar with handling COM components, programming can be done using function browsers or auto-completion that show you the available properties and functions/methods including their parameter names. It is thus often possible to use the available functions without consulting the documentation.

Since the syntax of calling COM components is a little different in every programming language, we focus in our examples on MATLAB code. MATLAB has a very generic syntax and it should be easy to adapt the code to any other language. Further examples are included in the software distribution. These examples get installed with the Photonic Application Suite and can usually be found at the following location:

*C:\Program Files\Keysight\Photonic Application Suite\Examples*

Here is a simple example which starts a measurement (similar to press the button “Run Single”):

```
% Connect to Engine Manager
EngineMgr=actxserver('AgServerILPDL.EngineMgr');

% List all Engines currently running
EngineIDs=EngineMgr.EngineIDs;

% Always connect to first engine
Engine=EngineMgr.OpenEngine(EngineIDs(1));

% Start measurement
Engine.StartMeasurement;

% Release the engine and the engine manager
Engine.release;
EngineMgr.release;
```

This example assumes that you have already started the Client Software and thus a server is already running. Therefore, this example connects to an existing engine.

The next chapter explains how to create a new instance of the engine, in case the server has not yet been started.

## Creating a New Engine

When writing your own software, it is very useful to start the Client Software first. Starting the Client Software will start the Server Software and creates an instance of the “Engine” COM component which handles the hardware communication. If your self-written software connects to this engine, you can see changes immediately in the client user interface. This is extremely useful in the debugging phase.

Finalizing your software, you probably don’t need the user interface anymore. In that case, you can start the server yourself and create an instance of the engine:

```
% Connect to Engine Manager
EngineMgr=actxserver('AgServerILPDL.EngineMgr');

% Create a new engine
Engine=EngineMgr.NewEngine;

% Load configuration file
Engine.LoadConfiguration('c:\\test.agconfig');

% Activate engine
Engine.Activate;

% Start measurement
Engine.StartMeasurement;

% Deactivate engine
Engine.DeActivate;

% Release the engine and the engine manager
Engine.release;
EngineMgr.release;
```

Note that you have to activate/deactivate the engine explicitly. If connecting to an existing engine, this is done by the client. When activating the engine, communication with the hardware is started.

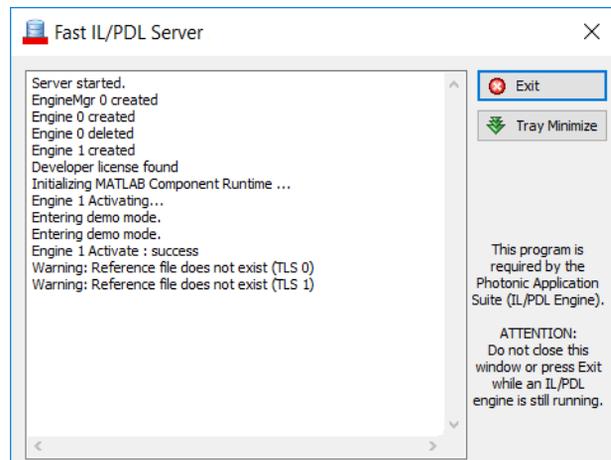
We recommend to configure your engine initially using the Client Software and store the configuration using the “Save Configuration” menu. This example uses the “LoadConfiguration” command to load an .agconfig file which has been stored before by the client.

**NOTE**

An icon appears in the Windows taskbar (on the bottom of your screen) when the server has started (i.e. the engine manager has been invoked):



Left-click on the icon to bring the server window to the front. The server will display a protocol line when an engine has been created or deleted:



## Connecting to an Existing Engine

Use the following code to connect to an existing engine:

```
% Connect to Engine Manager
EngineMgr=actxserver('AgServerILPDL.EngineMgr');

% List all Engines currently running
EngineIDs=EngineMgr.EngineIDs;

% Always connect to first engine
Engine=EngineMgr.OpenEngine(EngineIDs(1));

% Start measurement
Engine.StartMeasurement;

% Release the engine and the engine manager
Engine.release;
EngineMgr.release;
```

### NOTE

In most of the following MATLAB examples, we assume that the engine has been created and activated before.

---

## Performing a Measurement

A full MATLAB demo named “ILPDL\_Remote\_Demo.m” is included in the C:\Program Files\Keysight\Photonic Application Suite\Examples\DemoMATLAB folder. Please take a look at this demo file even if you use a different programming language. The demo file is easy to understand even if you are not familiar with MATLAB. It shows you how to start a measurement, wait for the measurement to be finished and read the result.

The following code starts a measurement and waits for the measurement to be finished. The result is stored as an OMR file.

```
% Start
Engine.StartMeasurement;

% Wait for measurement to be finished
while Engine.Busy; pause(1); end;

% Get result object
MeasurementResult = Engine.MeasurementResult;

% Save as OMR file
MeasurementResult.Write('c:\test.omr');

% Release measurement object
MeasurementResult.release;
```

## Accessing the Measurement Result

The measurement result is returned as reference to a file object. This file COM object represents an OMR file and wraps all methods to read and write these files (Interface: IOMRFile).

### NOTE

Refer to the *N7700 Photonic Application Suite User's Guide* for a reference of the common interfaces, e.g. IOMRFile, IOMRGraph and IOMRProperty.

Once the measurement has finished, you can request a reference to this object and manipulate it. Typically you want to store the file somewhere on your hard disk or you want to access the graphs.

Saving the file can be done using the Write-method:

```
% Save as OMR file
MeasurementResult.Write('c:\test.omr');
```

Accessing the graph data of the IL measurement is done like this:

```
% Access IL-graph
Graph = MeasurementResult.Graph('TLS0_RXTXAvgIL');
noChannels = Graph.noChannels;
dataPerCurve = Graph.dataPerCurve;
YData = reshape(Graph.YData, dataPerCurve, noChannels);
xStart = Graph.xStart;
xStep = Graph.xStep;
```

The graphs in the file have names. The name “TLS0\_RXTXAvgIL” corresponds to the IL measurement performed with the first (or only) TLS in the setup. Please refer to the “Reference” section for a list of available names.

The data are returned as a one-dimensional array, even if it contains more than one channel. The data for each channel are concatenated. The array starts with the first channel and continues with the next channels. The MATLAB command “reshape” converts this to a 2-dimensional array, a matrix. You can use the properties “noChannels” to get the total number of contained channels and the property “dataPerCurve” to find out how many values belong to a single channel.

**NOTE**

Using automation commands it is possible to start a new measurement engine as well as connect to an existing one, for example, one started from the Keysight Client GUI.

Certain automation operations, such as the Write method of the OMRFile interface or the FileSave method of the IEngine interface, lock the measurement object while accessing it. This might cause temporary failures in automation operations that are triggered from other instances connected to the same N7700 engine server.

This might especially be the case when using engine automation in addition to running the Keysight Client, which locks the OMRFile object in memory while updating measurement data in the GUI right after a measurement and when performing any automatic history save (see [Automatic History Saves](#) on page 80 to learn how to disable automatic saves).

To deal with this behavior, either implement retries or delays on affected operations or avoid connecting multiple instances to the same measurement engine server.

---

## Reference: Interface “IEngineMgr”

The IEngineMgr interface is invoked using the following PROGID to identify the COM-server:

```
AgServerILPDL.EngineMgr
```

MATLAB:

```
EngineMgr=actxserver('AgServerILPDL.EngineMgr');
```

property Version

Type-Library:

```
get: HRESULT Version([out, retval] BSTR* pVal)
```

MATLAB:

```
Version=EngineMgr.Version;
```

Shows information about revision and build date of the engine.

method IsVersionGreaterOrEqual

Type-Library:

```
HRESULT IsVersionGreaterOrEqual([in] BSTR  
VersionNumber, [out, retval] BOOL* pVal);
```

MATLAB:

```
Engine=EngineMgr.IsVersionGreaterOrEqual('1.1.0.1');
```

Performs a check, whether the engine revision is greater than or equal to the revision number provided as a parameter.

method NewEngine

Type-Library:

```
HRESULT NewEngine([out, retval] IEngine** aEngine);
```

MATLAB:

```
Engine=EngineMgr.NewEngine;
```

Tells the engine manager to create a new engine and returns the corresponding handle.

## method OpenEngine

Type-Library:

```
HRESULT OpenEngine([in] LONG EngineID, [out, retval] IEngine**
aEngine);
```

MATLAB:

```
Engine=EngineMgr.OpenEngine(EngineID);
```

Returns the handle to an existing engine. The engine ID is a unique identifier which can be obtained by requesting the property "EngineIDs".

## property EngineIDs

Type-Library:

```
get: HRESULT EngineIDs([out, retval] SAFEARRAY(LONG) * pVal)
```

MATLAB:

```
EngineIDs=EngineMgr.EngineIDs;
```

Returns the IDs of all created engines.

## method DeleteEngine

Type-Library:

```
HRESULT DeleteEngine([in] IEngine* aEngine);
```

MATLAB:

```
EngineMgr.DeleteEngine(Engine);
```

To ensure that all memory clean-up operations are executed properly, it is recommended to call the EngineMgr's DeleteEngine method with the current engine object as an argument after running the engine's DeActivate method and before releasing the engine object. After that, either create a new engine or release the EngineMgr object.

## Reference: Interface “IEngine”

The IEngine interface is invoked either using the method “NewEngine” or “OpenEngine” of the IEngineMgr interface:

MATLAB:

```
Engine=EngineMgr.NewEngine;
```

property Version

Type-Library:

```
get: HRESULT Version([out, retval] BSTR* pVal)
```

MATLAB:

```
Version=Engine.Version;
```

Shows information about revision and build date of the engine.

method IsVersionGreaterOrEqual

Type-Library:

```
HRESULT IsVersionGreaterOrEqual([in] BSTR  
VersionNumber, [out, retval] BOOL* pVal);
```

MATLAB:

```
Engine=Engine.IsVersionGreaterOrEqual('1.1.0.1');
```

Performs a check, whether the engine revision is greater than or equal to the revision number provided as a parameter.

method Activate

Type-Library:

```
HRESULT Activate(void);
```

MATLAB:

```
Engine.Activate;
```

Activates the engine. Usually a configuration file, created using the client software, should be loaded prior to activating the engine. This is done using the “LoadConfiguration” method. Activating an engine will cause the server to communicate with the instruments.

## method DeActivate

Type-Library:

```
HRESULT DeActivate(void);
```

MATLAB:

```
Engine.DeActivate;
```

Deactivates the engine. You should deactivate the engine before loading a different configuration.

**NOTE**

After deactivating an engine and before releasing the engine object and its corresponding EngineMgr object, the EngineMgr's DeleteEngine method should be called. Refer to [method DeleteEngine](#) on page 103 for further details.

---

## property Active

Type-Library:

```
get: HRESULT Active([out, retval] BOOL* pVal);
```

MATLAB:

```
x = Engine.Active;
```

Returns 0/1 if the engine is inactive/active.

## property Busy

Type-Library:

```
get: HRESULT Busy([out, retval] BOOL* pVal);
```

MATLAB:

```
x = Engine.Busy;
```

Returns 0/1 if the engine is not-busy/busy. The engine might be busy when transferring large amounts of data to or from the instruments, when being activated, or when it is waiting for a measurement operation to complete. This flag is intended for operations that run asynchronously, such as those mentioned above. Please note that the flag might not reflect the engine's actual busy state until an engine activation has been performed. Especially after creating an engine using an EngineMgr object and after loading a configuration file, but before executing the Activate method, the Busy flag will return 1 (busy).

## property EmulationMode

## Type-Library:

```
get: HRESULT EmulationMode([out, retval] BOOL* pVal);
put: HRESULT EmulationMode([in] BOOL Val);
```

## MATLAB:

```
b = Engine.EmulationMode;
Engine.EmulationMode = b;
```

Sets or gets the emulation (demo) mode setting. If operated in demo mode, no actual communication with instruments will be performed and simulated measurement results will be shown instead.

## method LoadConfiguration

## Type-Library:

```
HRESULT LoadConfiguration([in] BSTR Filename);
```

## MATLAB:

```
Engine.LoadConfiguration('c:\\test.agconfig');
```

Loads a configuration file (extension: .agconfig). The engine should be inactive while loading a new config file, i.e., either before calling Activate or after calling Deactivate.

## property Configuration

## Type-Library:

```
get: HRESULT Configuration([out, retval] BSTR* ConfigXML);
put: HRESULT Configuration([in] BSTR ConfigXML);
```

## MATLAB:

```
xml = Engine.Configuration;
Engine.Configuration = xml;
```

Transfers the contents of an .agconfig-file to or from the engine. An .agconfig-file contains XML formatted text.

## method StartReference

## Type-Library:

```
HRESULT StartReference(void);
```

MATLAB:

```
Engine.StartReference;
```

Starts a reference measurement in Auto Pattern mode (see [Performing a Reference Measurement](#) on page 51).

method StartCustomReference / StartCustomReferenceSinglePort

Type-Library:

```
HRESULT StartCustomReference([in] SAFEARRAY(LONG) Ports);
HRESULT StartCustomReferenceSinglePort([in] LONG Port);
```

MATLAB:

```
Engine.StartCustomReference(int32([3;6]));
Engine.StartCustomReferenceSinglePort(3);
```

Starts a reference measurement. As a parameter, an integer array of port numbers or a single port number, respectively (starting at 0 for port 1), must be provided. A reference sweep will be performed and reference data for all provided ports will be stored.

## NOTE

When using MATLAB (and potentially other automation environments), it might not be possible to match the argument type (integer array) when trying to perform this operation on a single port. In such a situation, use `Engine.StartCustomReferenceSinglePort(nPort)` instead.

method ClearReferenceChannels / ClearReferenceChannelsSinglePort

Type-Library:

```
HRESULT ClearReferenceChannels([in] SAFEARRAY(LONG) Ports);
HRESULT ClearReferenceChannelsSinglePort([in] LONG Port);
```

MATLAB:

```
Engine.ClearReferenceChannels(int32([3;6]));
Engine.ClearReferenceChannelsSinglePort(3);
```

Clears reference data from the currently loaded reference file. The saved file will not be changed by this operation. To update the saved reference file, the modified reference needs to be saved with the same filename. As a parameter, an integer array of port numbers or a single port number, respectively (starting at 0 for port 1), must be provided.

**NOTE**

When using MATLAB (and potentially other automation environments), it might not be possible to match the argument type (integer array) when trying to perform this operation on a single port. In such a situation, use `Engine.ClearReferenceChannelsSinglePort(nPort)` instead.

---

method `CopyReferenceChannel` / `CopyReferenceChannelSinglePort`

Type-Library:

```
HRESULT CopyReferenceChannel([in] SourcePort, [in]
SAFEARRAY(LONG) TargetPorts);
HRESULT CopyReferenceChannelSinglePort([in] SourcePort, [in]
LONG TargetPort);
```

MATLAB:

```
Engine.CopyReferenceChannel(5, int32([3;6]));
Engine.CopyReferenceChannelSinglePort(5, 3);
```

Copies reference data from the source port to all target ports in the currently loaded reference file. The saved file will not be changed by this operation. To update the saved reference file, the modified reference needs to be saved with the same filename. As parameters, the source port of the copy operation and an integer array of the target port numbers or a single port number, respectively (parameters starting at 0 for port 1), must be provided.

**NOTE**

When using MATLAB (and potentially other automation environments), it might not be possible to match the argument type (integer array) when trying to perform this operation on a single port. In such a situation, use `Engine.CopyReferenceChannelSinglePort(nPort)` instead.

---

method `UndoReferenceOperation`

Type-Library:

```
HRESULT UndoReferenceOperation(void);
```

MATLAB:

```
Engine.UndoReferenceOperation;
```

Reverts the currently loaded reference to the state prior to the last reference operation.

property ReferenceList

Type-Library:

```
get: HRESULT ReferenceList([out, retval] BSTR* pVal);
```

MATLAB:

```
s = Engine.ReferenceList;
```

Returns an xml string containing the filenames and important parameters for all reference files available on the engine server.

method DeleteReferenceFile

Type-Library:

```
HRESULT DeleteReferenceFile([in] BSTR Filename);
```

MATLAB:

```
Engine.DeleteReferenceFile('Reference.pbin');
```

Deletes a reference file on the engine server. Use `Engine.ReferenceList` to get a list of reference files present on the server.

method ValidateSettings

Type-Library:

```
HRESULT ValidateSettings(void);
```

MATLAB:

```
Engine.ValidateSettings;
```

Checks measurement parameters for validity and prompts for adjustments, if required, e.g. wavelength range exceeding the sweep range of the TLS used. Will also check whether the parameters match the currently selected reference.

method ValidateSettingsNoRefCheck

Type-Library:

```
HRESULT ValidateSettingsnoRefCheck(void);
```

MATLAB:

```
Engine.ValidateSettingsnoRefCheck;
```

Checks measurement parameters for validity and prompts for adjustments, if required, e.g. wavelength range exceeding the sweep range of the TLS used. Will not check whether the parameters match the current reference. Can be used before taking a new reference with changed parameters.

property DarkIncludeMeasurement

Type-Library:

```
get: HRESULT DarkIncludeMeasurement([out, retval] BOOL* pVal);
put: HRESULT DarkIncludeMeasurement([in] BOOL Val);
```

MATLAB:

```
b = Engine.DarkIncludeMeasurement;
Engine.DarkIncludeMeasurement = b;
```

Defines, whether dark current measurements should be performed in addition to the swept measurement.

property ERIncludeMeasurement

Type-Library:

```
get: HRESULT ERIncludeMeasurement([out, retval] BOOL* pVal);
put: HRESULT ERIncludeMeasurement([in] BOOL Val);
```

MATLAB:

```
b = Engine.ERIncludeMeasurement;
Engine.ERIncludeMeasurement = b;
```

Defines, whether stepped Polarization Extinction Ratio measurements should be performed in addition to the swept measurement.

property HighCapacitanceMode

Type-Library:

```
get: HRESULT HighCapacitanceMode([out, retval] BOOL* pVal);
put: HRESULT HighCapacitanceMode([in] BOOL Val);
```

MATLAB:

```
b = Engine.HighCapacitanceMode;
Engine.HighCapacitanceMode = b;
```

Defines, whether engine operates in regular or in high capacitance mode (refer to section [Instrument Setup Parameters](#) on page 43 for further details).

method StartMeasurement

Type-Library:

```
HRESULT StartMeasurement(void);
```

MATLAB:

```
Engine.StartMeasurement;
```

Starts a measurement.

method StartMeasurementRepeat

Type-Library:

```
HRESULT StartMeasurementRepeat(void);
```

MATLAB:

```
Engine.StartMeasurementRepeat;
```

Starts a series of measurements. Will stop after StopMeasurement method has been called. Will not save any measurement data automatically, except for any automatic history saves that might have been configured in an engine client (IL/PDL Engine GUI), connected to the same engine server.

method StopMeasurement

Type-Library:

```
HRESULT StopMeasurement(void);
```

MATLAB:

```
Engine.StopMeasurement;
```

During a measurement you can abort the current measurement sweep and stop the repeat operation by using this method. The measurement will be stopped as soon as possible, but usually not immediately, because some hardware operations have to be completed first. Usually the next measurement is completed before stopping, so this method should be used primarily for stopping the Repeat mode operation.

method FileSave

Type-Library:

```
HRESULT FileSave([in] BSTR Filename);
```

MATLAB:

```
Engine.FileSave('test.omr');
```

Saves the measurement result to an OMR file at the location provided as parameter.

See the last Note in [Accessing the Measurement Result](#) on page 100.

property MeasurementResult

Type-Library:

```
HRESULT MeasurementResult([out, retval] IOMRFile** pVal);
```

MATLAB:

```
Result = Engine.MeasurementResult;
```

Returns a reference to the underlying IOMRFile object which contains the measurement result.

## NOTE

Refer to the *N7700 Photonic Application Suite User's Guide* for a reference of the common interfaces, e.g. IOMRFile, IOMRGraph and IOMRProperty.

---

See the last Note in [Accessing the Measurement Result](#) on page 100.

property ResolutionValues

Type-Library:

```
get: HRESULT ResolutionValues([out, retval]  
SAFEARRAY(double)* pVal);
```

MATLAB:

```
StringArray = Engine.ResolutionValues;
```

Returns a double array containing the suggested window widths for the moving-average smoothing in nm.

## property Resolution

## Type-Library:

```
get: HRESULT Resolution([out, retval] DOUBLE* pVal);
put: HRESULT Resolution([in] DOUBLE Val);
```

## MATLAB:

```
d = Engine.Resolution;
Engine.Resolution = d;
```

Sets or gets the window widths for the moving-average smoothing in nm.

## method ReCalculate

## Type-Library:

```
HRESULT ReCalculate(void);
```

## MATLAB:

```
Engine.ReCalculate;
```

Triggers a recalculation. Call this method if you have changed evaluation parameters such as resolution or, e.g., GenerateTETMData.

## property MaxAllowedRefLoss

## Type-Library:

```
get: HRESULT MaxAllowedRefLoss([out, retval] DOUBLE* pVal);
put: HRESULT MaxAllowedRefLoss([in] DOUBLE Val);
```

## MATLAB:

```
d = Engine.MaxAllowedRefLoss;
Engine.MaxAllowedRefLoss = d;
```

Sets or gets the maximum allowed reference loss in dB. If the minimum difference between the set TLS power and the measured power at the used power / current meter channel(s) during a reference sweep is larger than this value, the engine will display a warning. To avoid warnings, increase this value, if you are using additional components in the reference measurement path that contribute to the overall loss.

## property PWMCount

## Type-Library:

```
get: HRESULT PWMCount([out, retval] LONG* pVal);
```

MATLAB:

```
NumberOfPWMSAttached = Engine.PWMCount;
```

Returns the number of power / current meters configured in the current setup. Will only return a positive, i.e., valid, value after engine activation has been started.

method GetPWMChannelCount

Type-Library:

```
HRESULT GetPWMChannelCount([in] LONG nPWM, [out, retval] LONG* pVal);
```

MATLAB:

```
PWMChannelCount = Engine.GetPWMChannelCount(nPWM);
```

Returns the number of ports for a given power/ current meter in the setup. You can obtain the number of power / current meters attached and configured by using Engine.GetPWMChannelCount.

property TLSCount

Type-Library:

```
get: HRESULT TLSCount([out, retval] LONG* pVal);
```

MATLAB:

```
NumberOfTLSAttached = Engine.TLSCount;
```

Returns the number of TLS configured in the current setup. Will only return a non-zero, i.e., valid, value after engine activation has been started.

method GetTLSInterface

Type-Library:

```
HRESULT GetTLSInterface([in] LONG nTLS, [out, retval] IEngineTLS** pTLSInterface);
```

MATLAB:

```
TLS=Engine.GetTLSInterface(nTLS);
```

Returns the handle to an existing TLS object. This TLS object is used to set and query TLS specific measurement parameters. Refer to [Reference: Interface "IEngineTLS"](#) on page 135 for details on those settings. Will only return a valid object after engine activation has been started.

## property KeepRawData

## Type-Library:

```
get: HRESULT KeepRawData([out, retval] BOOL* pVal);
put: HRESULT KeepRawData([in] BOOL Val);
```

## MATLAB:

```
b = Engine.KeepRawData;
Engine.KeepRawData = b;
```

Defines, whether stored measurement files will contain measurement raw data. If measurement raw data is stored, files can be re-evaluated at a later time.

## property PWMAutoRanging

## Type-Library:

```
get: HRESULT PWMAutoRanging([out, retval] BOOL* pVal);
put: HRESULT PWMAutoRanging([in] BOOL Val);
```

## MATLAB:

```
b = Engine.PWMAutoRanging;
Engine.PWMAutoRanging = b;
```

Defines, whether the power ranges of attached power / current meters (and return loss modules, if present) are adjusted automatically to avoid overrange and underrange measurements.

## property MPPMAutoGain

## Type- Library:

```
get: HRESULT MPPMAutoGain([out, retval] BOOL* pVal);
put: HRESULT MPPMAutoGain([in] BOOL Val);
```

## MATLAB:

```
b= Engine.MPPMAutoGain;
Engine.MPPMAutoGain= b;
```

If set to true, the measurement engine will configure any MPPM to use its auto gain feature, which gives improved dynamic range performance. Refer to the [Instrument Setup Parameters](#) on page 43 for further details.

## property TLSExtendedSweepRange

Type- Library:

```
get: HRESULT TLSExtendedSweepRange([out, retval] BOOL* pVal);
```

```
put: HRESULT TLSExtendedSweepRange([in] BOOL Val);
```

MATLAB:

```
b= Engine.TLSExtendedSweepRange;
```

```
Engine.TLSExtendedSweepRange= b;
```

If set to true, power meter sampling will start sooner after the TLS sweep start.

Refer to the [Instrument Setup Parameters](#) on page 43 for further details.

## property GenerateTETMData

Type-Library:

```
get: HRESULT GenerateTETMData([out, retval] BOOL* pVal);
```

```
put: HRESULT GenerateTETMData([in] BOOL Val);
```

MATLAB:

```
b = Engine.GenerateTETMData;
```

```
Engine.GenerateTETMData = b;
```

Defines whether TE/TM and Mueller data is computed during measurement evaluation.

## property GenerateReturnLossData

Type-Library:

```
get: HRESULT GenerateReturnLossData([out, retval] BOOL* pVal);
```

```
put: HRESULT GenerateReturnLossData([in] BOOL Val);
```

MATLAB:

```
b = Engine.GenerateReturnLossData;
```

```
Engine.GenerateReturnLossData = b;
```

Defines whether return loss data is computed during measurement evaluation (if a return loss module is connected and configured for the current setup).

property ReturnLossModulePresent

Type-Library:

```
get: HRESULT ReturnLossModulePresent([out, retval] BOOL*
pVal);
```

MATLAB:

```
x = Engine.ReturnLossModulePresent;
```

Returns 1 if a return loss module is used in the current setup or 0 if not.

property GenerateResponsivityData

Type-Library:

```
get: HRESULT GenerateResponsivityData([out, retval] BOOL*
pVal);
```

```
put: HRESULT GenerateResponsivityData([in] BOOL Val);
```

MATLAB:

```
b = Engine.GenerateResponsivityData;
```

```
Engine.GenerateResponsivityData = b;
```

Defines whether photodiode responsivity data is computed during measurement evaluation (if a power / current meter instrument with electrical inputs is connected and configured for the current setup).

property GenerateDiodeCurrentData

Type-Library:

```
get: HRESULT GenerateDiodeCurrentData([out, retval] BOOL*
pVal);
```

```
put: HRESULT GenerateDiodeCurrentData([in] BOOL Val);
```

MATLAB:

```
b = Engine.GenerateDiodeCurrentData;
```

```
Engine.GenerateDiodeCurrentData = b;
```

Defines whether photodiode current data is computed during measurement evaluation (if a power / current meter instrument with electrical inputs is connected and configured for the current setup).

property ElectricalInputsPresent

Type-Library:

```
get: HRESULT ElectricalInputsPresent([out, retval] BOOL* pVal);
```

MATLAB:

```
x = Engine.ElectricalInputsPresent;
```

Returns 1 if a power / current meter with electrical inputs is used in the current setup or 0 if not.

method IsBiasEnabled

Type-Library:

```
HRESULT IsBiasEnabled([in] LONG nPWM, [in] LONG nChannel, [out, retval] BOOL* pVal);
```

MATLAB:

```
b = Engine.IsBiasEnabled(nPWM, nChannel);
```

Checks whether a given port of a given power / current meter has an electrical input that allows for bias adjustment.

Refer to [property PWMCount](#) on page 113 and [method GetPWMChannelCount](#) on page 114 for input parameters nPWM and nChannel.

method GetBias

Type-Library:

```
HRESULT GetBias([in] LONG nPWM, [in] LONG nChannel, [out, retval] DOUBLE* pVal);
```

MATLAB:

```
Bias = Engine.GetBias(nPWM, nChannel);
```

Returns the current bias setting for a given channel of a given power / current meter. Use `Engine.IsBiasEnabled(nPWM, nChannel)` to check whether that port supports bias adjustment.

Refer to [property PWMCount](#) on page 113 and [method GetPWMChannelCount](#) on page 114 for input parameters nPWM and nChannel.

method SetBias

Type-Library:

```
HRESULT SetBias([in] LONG nPWM, [in] LONG nChannel, [in]
DOUBLE Bias);
```

MATLAB:

```
Engine.SetBias(nPWM, nChannel, Bias);
```

Sets the bias setting for a given channel of a given power / current meter. Use `Engine.IsBiasEnabled(nPWM, nChannel)` to check whether that port supports bias adjustment.

Refer to [property PWMCount](#) on page 113 and [method GetPWMChannelCount](#) on page 114 for input parameters `nPWM` and `nChannel`.

#### property BiasMode

Type-Library:

```
get: HRESULT Bias([out, retval] LONG* pVal);
put: HRESULT Bias([in] LONG Val);
```

MATLAB:

```
i= Engine.Bias;
Engine.Bias = i;
```

Sets or gets the behavior for bias voltage application. Set to 0/1/2 for Off/On/Auto. Refer to section [Bias Settings](#) on page 88 for further details.

#### method ZeroAllMPPMs

Type-Library:

```
HRESULT ZeroAllMPPMs(void);
```

MATLAB:

```
Engine.ZeroAllMPPMs;
```

Starts the zeroing procedure on all configured power / current meters (and return loss modules, if present) in the current setup.

#### property LambdaZeroingMode

Type-Library:

```
get: HRESULT LambdaZeroingMode([out, retval] LONG* pVal);
put: HRESULT LambdaZeroingMode([in] LONG Val);
```

MATLAB:

```
i= Engine.LambdaZeroingMode;
Engine.LambdaZeroingMode = i;
```

Sets or gets the behavior when a lambda zeroing operation is suggested by the software. Set to 0/1/2 for Automatically/Always Ask/Manual Only. Refer to [Instrument Setup Parameters](#) on page 43 for further details.

#### method ZeroAllTLS

Type-Library:

```
HRESULT ZeroAllTLS(void);
```

MATLAB:

```
Engine.ZeroAllTLS;
```

Performs a lambda zeroing operation on all tunable laser sources used in the current configuration.

#### method CalibrateRLM

Type-Library:

```
HRESULT CalibrateRLM(void);
```

MATLAB:

```
Engine.CalibrateRLM;
```

Starts the return loss module calibration procedure (if a return loss module is present in the current setup). Use `Engine.RLMCalReflectorValue` to enter the correct value for your reference reflection artifact.

#### property RLMCalReflectorValue

Type-Library:

```
get: HRESULT RLMCalReflectorValue([out, retval] DOUBLE* pVal);
```

```
put: HRESULT RLMCalReflectorValue([in] DOUBLE Val);
```

MATLAB:

```
d = Engine.RLMCalReflectorValue;
```

```
Engine.RLMCalReflectorValue = d;
```

Sets or gets the return loss of the reference reflection artifact used in return loss calibration in dB.

## property EventPropertiesChanged

## Type-Library:

```
get: HRESULT EventPropertiesChanged([out, retval] LONG* pVal);
```

## MATLAB:

```
nEvent = Engine.EventPropertiesChanged;
```

This property is a 32 bit integer value. The value will be increased whenever the state of the engine has changed. By polling this value and observing the changes, a client can be notified of any change of the engine state.

## property EventMeasurementFinished

## Type-Library:

```
get: HRESULT EventMeasurementFinished([out, retval] LONG* pVal);
```

## MATLAB:

```
nEvent = Engine.EventMeasurementFinished;
```

This property is a 32 bit integer value. The value will be increased whenever the engine has finished a measurement. By polling this value and observing the changes, a client can be notified of any change of the engine state.

## property EventReferenceOperationFinished

## Type-Library:

```
get: HRESULT EventReferenceOperationFinished([out, retval] LONG* pVal);
```

## MATLAB:

```
nEvent = Engine.EventReferenceOperationFinished;
```

This property is a 32 bit integer value. The value will be increased whenever the engine has finished a reference operation, e.g. performing a reference sweep, deleting the reference data for certain ports, etc.. By polling this value and observing the changes, a client can be notified of any change of the engine state.

## property ProtocolText

## Type-Library:

```
get: HRESULT ProtocolText([out, retval] BSTR* pVal)
```

MATLAB:

```
ProtocolText = Engine.ProtocolText;
```

Returns a string containing all engine messages separated by a '|' character. If the number of messages exceeds 1000, the oldest message is deleted from the list for each new message.

property ProtocolMin

Type-Library:

```
get: HRESULT ProtocolMin([out, retval] LONG* pVal)
```

MATLAB:

```
First = Engine.ProtocolMin;
```

Returns the index to the first entry in the message list.

property ProtocolMax

Type-Library:

```
get: HRESULT ProtocolMax([out, retval] LONG* pVal)
```

MATLAB:

```
Last = Engine.ProtocolMax;
```

Returns the index to the most recent entry in the message list.

method GetProtocolTextAt

Type-Library:

```
HRESULT GetProtocolTextAt([in] LONG nMin, [in] LONG nMax,
[out, retval] BSTR* pVal);
```

MATLAB:

```
s = Engine.GetProtocolTextAt(Engine.ProtocolMin,
Engine.ProtocolMax);
```

Returns a portion of the message history defined by nMin und nMax.

property UserInputWaiting

Type-Library:

```
get: HRESULT UserInputWaiting([out, retval] BOOL* pVal);
```

```
put: HRESULT UserInputWaiting([in] BOOL Val);
```

MATLAB:

```
b = Engine.UserInputWaiting;
Engine.UserInputWaiting = b;
```

Returns 1 if the engine is waiting for a user input, e.g. to choose between different options or to acknowledge an error message. Returns 0 otherwise. Set to 0 if you have handled the user input (see [method UserInputResponse](#) on page 123).

property UserInputPrompt

Type-Library:

```
get: HRESULT UserInputPrompt([out, retval] BSTR* pVal)
```

MATLAB:

```
Prompt = Engine.UserInputPrompt;
```

Returns a string containing the prompt of the current user input request, if any.

property UserInputChoice

Type-Library:

```
get: HRESULT UserInputChoice([out, retval] BSTR* pVal)
```

MATLAB:

```
Choices = Engine.UserInputChoice;
```

Returns a string containing all possible responses to the user input requests separated by a '|' character. Each response contains a corresponding number to be used by `Engine.UserInputResponse` and the description of the response, e.g. for labeling a button. Number and description are separated by a ',' character.

method UserInputResponse

Type-Library:

```
HRESULT UserInputResponse([in] LONG nResponse);
```

MATLAB:

```
Engine.UserInputResponse(1);
```

Set the response to a user input request. Must be a number from the list obtained by `Engine.UserInputChoice`. Set `Engine.UserInputWaiting` to 0 after setting `Engine.UserInputResponse` for the engine to continue operation.

method `DataLoggingTimeout`

Type-Library:

```
get: HRESULT DataLoggingTimeout([out, retval] DOUBLE* pVal);
put: HRESULT DataLoggingTimeout([in] DOUBLE Val);
```

MATLAB:

```
d = Engine.DataLoggingTimeout;
Engine.DataLoggingTimeout = d;
```

Sets or gets the data logging timeout value in minutes (default: 5.0 min). This is the amount of time the engine will wait for measurement data from the instruments before declaring the measurement failed.

property `ImmediateEvaluation`

Type-Library:

```
get: HRESULT ImmediateEvaluation ([out, retval] BOOL* pVal);
put: HRESULT ImmediateEvaluation ([in] BOOL Val);
```

MATLAB:

```
b = Engine.ImmediateEvaluation;
Engine.ImmediateEvaluation= b;
```

If set to true, measurement data will be evaluated right after data acquisition. If set to false, re-evaluation can be performed at a later time, if raw data is stored within files.

Refer to the [Advanced Parameters](#) on page 47 for further details.

property `QuickPolarimeterSetup`

Type-Library:

```
get: HRESULT QuickPolarimeterSetup ([out, retval] BOOL*
pVal);
put: HRESULT QuickPolarimeterSetup ([in] BOOL Val);
```

MATLAB:

```
b = Engine.QuickPolarimeterSetup;
Engine.QuickPolarimeterSetup= b;
```

If set to true, measurement initialization time is reduced.

Refer to the [Advanced Parameters](#) on page 47 for further details.

property MeasurementSpeedLegacyMode

Type-Library:

```
get: HRESULT MeasurementSpeedLegacyMode ([out, retval] BOOL*
pVal);
put: HRESULT MeasurementSpeedLegacyMode ([in] BOOL Val);
```

MATLAB:

```
b = Engine.MeasurementSpeedLegacyMode;
Engine.MeasurementSpeedLegacyMode= b;
```

If set to false, measurement initialization time is reduced.

Refer to the [Advanced Parameters](#) on page 47 for further details.

property SOPSearchQuickMode

Type-Library:

```
get: HRESULT SOPSearchQuickMode ([out, retval] BOOL* pVal);
put: HRESULT SOPSearchQuickMode ([in] BOOL Val);
```

MATLAB:

```
b = Engine.SOPSearchQuickMode;
Engine.SOPSearchQuickMode = b;
```

If set to true, reference measurement initialization time is reduced.

Refer to the [Advanced Parameters](#) on page 47 for further details.

method ForceMediumN7786BInitialization

Type-Library:

```
HRESULT ForceMediumN7786BInitialization (void);
```

MATLAB:

```
Engine.ForceMediumN7786BInitialization;
```

Ensures that certain N7786B initialization steps will be performed before the next measurement.

Refer to the [Advanced Parameters](#) on page 47 for further details.

method ForceFullN7786BInitialization

Type-Library:

```
HRESULT ForceFullN7786BInitialization (void);
```

MATLAB:

```
Engine.ForceFullN7786BInitialization;
```

Ensures that all N7786B initialization steps will be performed before the next measurement.

Refer to the [Advanced Parameters](#) on page 47 for further details.

method ForceTLSInitialization

Type-Library:

```
HRESULT ForceTLSInitialization (void);
```

MATLAB:

```
Engine.ForceTLSInitialization;
```

Ensures that all TLS initialization steps will be performed before the next measurement.

Refer to the [Advanced Parameters](#) on page 47 for further details.

method StartILDeembeddingMasterPortMeasurement

Type-Library:

```
HRESULT StartILDeembeddingMasterPortMeasurement (void);
```

MATLAB:

```
Engine.StartILDeembeddingMasterPortMeasurement;
```

Starts an IL de-embedding master port measurement.

Refer to **Performing Master Port Measurements** on page 63 for details.

method StartILDeembeddingSpecificPathMeasurement

Type-Library:

```
HRESULT StartILDeembeddingSpecificPathMeasurement([in] BSTR
Filename, BOOL Overwrite);
```

MATLAB:

```
Engine.StartILDeembeddingSpecificPathMeasurement(Filename,
1);
```

Starts an IL de-embedding specific path measurement and saves resulting OMR file using the provided file name. The `overwrite` argument is used to specify whether an existing file is overwritten automatically. Otherwise, there will be an error message and a user input prompt that must be handled accordingly, in case the target file exists. The `Filename` argument must contain extension (.omr), since this is not added automatically if omitted.

Refer to [Performing Specific Path Measurements](#) on page 63 for details.

method GetILDeembeddingFileList

Type-Library:

```
HRESULT GetILDeembeddingFileList([in] LONG
nPort, [out,retval] BSTR*pVal);
```

MATLAB:

```
ILDeembeddingFileList =
Engine.GetILDeembeddingFileList(nPort);
```

Returns a string containing all files to be used for IL de-embedding on the port defined by `nPort` (starting at 0 for port 1). File / path names are considered relative to value defined by the “[property ILDeembeddingTargetFolder](#)”.

Refer to [Applying IL De-embedding Data](#) on page 64 for details.

method SetILDeembeddingFileList

Type-Library:

```
HRESULT SetILDeembeddingFileList([in] LONG nPort, BSTR
ILDeembeddingFileList);
```

MATLAB:

```
Engine.SetILDeembeddingFileList(nPort,
ILDeembeddingFileList);
```

Sets a string defining all the files to be used for IL de-embedding on the port defined by `nPort` (starting at 0 for port 1). File / path names are considered relative to value defined by "[property ILDeembeddingTargetFolder](#)".

Refer to [Applying IL De-embedding Data](#) on page 64 for further details, especially on how to combine multiple files that are to be applied on a single port and whether each one should be added to the overall correction data in an additive or subtractive manner.

property ILDeembeddingTargetFolder

Type-Library:

```
get: HRESULT ILDeembeddingTargetFolder([out, retval] BSTR* pVal);
```

```
put: HRESULT ILDeembeddingTargetFolder([in] BSTR TargetFolder);
```

MATLAB:

```
s = Engine.ILDeembeddingTargetFolder;
Engine.ILDeembeddingTargetFolder = s;
```

Defines the default folder for IL de-embedding data files.

Refer to [Applying IL De-embedding Data](#) on page 64 for details.

property EnableILDeembedding

Type-Library:

```
get: HRESULT EnableILDeembedding([out, retval] BOOL* pVal);
```

```
put: HRESULT EnableILDeembedding([in] BOOL Val);
```

MATLAB:

```
b = Engine.EnableILDeembedding;
Engine.EnableILDeembedding = b;
```

Defines whether IL de-embedding data will be applied to subsequent measurements (or recalculations).

Refer to [Characterizing Components / Obtaining IL De-embedding Data](#) on page 62 for details.

## method SetStatic

Type-Library:

```
HRESULT SetStatic(void);
```

MATLAB:

```
Engine.SetStatic;
```

Performs a Static Mode operation, based on the “[property StaticWavelength](#)”, “[property StaticTLPower](#)”, “[property StaticPolarizationMode](#)” and corresponding other properties. For all the modes that use Stokes parameters as target values, a polarization stabilizer operation is started on the N7786B. Use “[method ReadCurrentStabilizerState](#)” to check the current stabilizer state and “[method StopStabilizer](#)” to stop stabilizer operation.

Refer to [Static Mode](#) on page 71 for details.

## method StopStabilizer

Type-Library:

```
HRESULT StopStabilizer(void);
```

MATLAB:

```
Engine.StopStabilizer;
```

Stops the N7786B stabilizing operation, in case it is currently active. Use “[method ReadCurrentStabilizerState](#)” to check the current stabilizer state.

Refer to [Static Mode](#) on page 71 for details.

## property StaticWavelength

Type-Library:

```
get: HRESULT StaticWavelength([out, retval] DOUBLE* pVal);
```

```
put: HRESULT StaticWavelength([in] DOUBLE Val);
```

MATLAB:

```
d = Engine.StaticWavelength;
```

```
Engine.StaticWavelength = d;
```

Sets or gets the static mode wavelength in nm.

Refer to [Static Mode](#) on page 71 for details.

## property StaticTLSPower

Type-Library:

```
get: HRESULT StaticTLSPower([out, retval] DOUBLE* pVal);
```

```
put: HRESULT StaticTLSPower([in] DOUBLE Val);
```

MATLAB:

```
d = Engine.StaticTLSPower;
```

```
Engine.StaticTLSPower = d;
```

Sets or gets the static mode laser output power in dBm.

Refer to [Static Mode](#) on page 71 for details.

## property StaticPWMPort

Type-Library:

```
get: HRESULT StaticPWMPort([out, retval] LONG* pVal);
```

```
put: HRESULT StaticPWMPort([in] LONG Val);
```

MATLAB:

```
i = Engine.StaticPWMPort;
```

```
Engine.StaticPWMPort = i;
```

Sets or gets the power meter port to be considered for certain Static Mode modes, such as Last Measurement or OMR File.

Refer to [Static Mode](#) on page 71 for details.

## property StaticPolarizationMode

Type-Library:

```
get: HRESULT StaticPolarizationMode([out, retval]
eILPDLStatPolMode_t* pVal);
```

```
put: HRESULT StaticPolarizationMode([in] eILPDLStatPolMode_t
Val);
```

MATLAB:

```
mode = Engine.StaticPolarizationMode;
```

```
Engine.StaticPolarizationMode = mode;
```

```
Engine.StaticPolarizationMode =
'eILPDLStatPolModeStokesParameters';
```

```
Engine.StaticPolarizationMode = 4;
```

Enumerator names and values for eILPDLStatPolMode\_t:

eILPDLStatPolModeMeasurementMaxMin	0
eILPDLStatPolModeMeasurementPSP	1
eILPDLStatPolModeFileMaxMin	2
eILPDLStatPolModeFilePSP	3
eILPDLStatPolModeFilePSP	4
eILPDLStatPolModeWaveplateOrientations	5

Sets or gets the Static Mode operation to be performed, once “method **SetStatic**” is executed.

Refer to [Static Mode Modes](#) on page 73 for details.

property StaticPolarizationMaxMinMode

Type-Library:

```
get: HRESULT StaticPolarizationMaxMinMode([out, retval]
eILPDLStatPolMaxMinMode_t* pVal);
```

```
put: HRESULT StaticPolarizationMaxMinMode([in]
eILPDLStatPolMaxMinMode_t Val);
```

MATLAB:

```
mode = Engine.StaticPolarizationMaxMinMode;
```

```
Engine.StaticPolarizationMaxMinMode = mode;
```

```
Engine.StaticPolarizationMaxMinMode = 'eILPDLStatPolMax';
```

```
Engine.StaticPolarizationMaxMinMode = 0;
```

Enumerator names and values for eILPDLStatPolMaxMinMode\_t:

eILPDLStatPolMax	0
eILPDLStatPolMin	1

Defines whether to optimize for maximum or minimum transmission when executing “method `SetStatic`” with “property `StaticPolarizationMode`” set to `eILPDLStatPolModeMeasurementMaxMin` Or `eILPDLStatPolModeFileMaxMin`.

Refer to [Static Mode Modes](#) on page 73 for details.

property `StaticPolarizationPSPMode`

Type-Library:

```
get: HRESULT StaticPolarizationPSPMode([out, retval]
eILPDLStatPolPSPMode_t* pVal);
```

```
put: HRESULT StaticPolarizationPSPMode([in]
eILPDLStatPolPSPMode_t Val);
```

MATLAB:

```
mode = Engine.StaticPolarizationPSPMode;
Engine.StaticPolarizationPSPMode = mode;
Engine.StaticPolarizationPSPMode = ' eILPDLStatPolPSP1;
Engine.StaticPolarizationPSPMode = 0;
```

Enumerator names and values for `eILPDLStatPolPSPMode_t`:

<code>eILPDLStatPolPSP1</code>	0
<code>eILPDLStatPolPSP2</code>	1

Defines to which principal state of polarization (regarding PDL) the polarization is aligned when executing “method `SetStatic`” with “property `StaticPolarizationMode`” set to `eILPDLStatPolModeMeasurementPSP` Or `eILPDLStatPolModeFilePSP`.

Refer to [Static Mode Modes](#) on page 73 for details.

property `StaticFilename`

Type-Library:

```
get: HRESULT StaticFilename([out, retval] BSTR* pVal);
```

```
put: HRESULT StaticFilename([in] BSTR TargetFolder);
```

MATLAB:

```
s = Engine.StaticFilename;
```

```
Engine.StaticFilename = s;
```

Defines the file to be used when executing “method **SetStatic**” with “property **StaticPolarizationMode**” set to **eILPDLStatPolModeFileMaxMin** OR **eILPDLStatPolModeFilePSP**.

Refer to **Static Mode** on page 71 for details.

#### property StaticStokesParameters

Type-Library:

```
get: HRESULT StaticStokesParameters([out, retval]
SAFEARRAY(double) * pVal)
```

```
put: HRESULT StaticStokesParameters([in] SAFEARRAY(double)
newVal)
```

MATLAB:

```
d_array = Engine.StaticStokesParameters;
Engine.StaticStokesParameters = d_array;
```

Sets or gets the target normalized Stokes parameters (3-element double array) to be used when executing “method **SetStatic**” with “property **StaticPolarizationMode**” set to **eILPDLStatPolModeStokesParameters**.

Refer to **Static Mode Modes** on page 73 for details.

#### property StaticWaveplateOrientations

Type-Library:

```
get: HRESULT StaticWaveplateOrientations([out, retval]
SAFEARRAY(double) * pVal)
```

```
put: HRESULT StaticWaveplateOrientations([in]
SAFEARRAY(double) newVal)
```

MATLAB:

```
d_array = Engine.StaticWaveplateOrientations;
Engine.StaticWaveplateOrientations = d_array;
```

Sets or gets the target waveplate orientations in degrees (5-element double array) to be used when executing “method **SetStatic**” with “property **StaticPolarizationMode**” set to **eILPDLStatPolModeWaveplateOrientations**.

Refer to **Static Mode Modes** on page 73 for details.

method ReadCurrentStokesParameters

Type-Library:

```
get: HRESULT ReadCurrentStokesParameters([out, retval]
SAFEARRAY(double)* pVal)
```

MATLAB:

```
d_array = Engine.ReadCurrentStokesParameters();
```

Gets the current Stokes parameters (3-element double array), as measured by the N7786B polarimeter at the time of executing this method.

Refer to [Current Value Operations](#) on page 77 for details.

property ReadCurrentWaveplateOrientations

Type-Library:

```
get: HRESULT ReadCurrentWaveplateOrientations([out, retval]
SAFEARRAY(double)* pVal)
```

MATLAB:

```
d_array = Engine.ReadCurrentWaveplateOrientations;
```

Gets the current waveplate orientations in degrees (5-element double array), as currently set at the N7786B polarization-controller at the time of executing this method.

Refer to [Current Value Operations](#) on page 77 for details.

method ReadCurrentStabilizerState

Type-Library:

```
get: HRESULT ReadCurrentStabilizerState([out, retval] BOOL*
pVal)
```

MATLAB:

```
b = Engine.ReadCurrentStabilizerState();
```

Gets the N7786B polarization stabilizer state at the time of executing this method.

Refer to [Current Value Operations](#) on page 77 for details.

## Reference: Interface “IEngineTLS”

The IEngineTLS interface is returned by the method

```
TLS = IEngine.GetTLSInterface(0)
```

It provides access to the TLS related setup parameters in a multi-TLS setup and is also used with a single-TLS configuration.

### property ModelCode

Type-Library:

```
get: HRESULT ModelCode([out, retval] BSTR* pVal);
```

MATLAB:

```
s = TLS.ModelCode;
```

Gets the model code of the current TLS.

### property Bands

Type-Library:

```
get: HRESULT Bands([out, retval] BSTR* pVal);
```

MATLAB:

```
s = TLS.Bands;
```

Gets the wavelength bands covered by the current TLS. The returned string is a concatenation of the band characters, e.g. “CL”.

### property Include

Type-Library:

```
get: HRESULT Include([out, retval] BOOL* pVal);
```

```
put: HRESULT Include([in] BOOL Val);
```

MATLAB:

```
b = TLS.Include;
```

```
TLS.Include = b;
```

Includes or excludes this TLS from subsequent operations (Taking references, running measurements, performing return loss calibrations)

## property WavelengthStart

## Type-Library:

```
get: HRESULT WavelengthStart([out, retval] DOUBLE* pVal);
put: HRESULT WavelengthStart([in] DOUBLE Val);
```

## MATLAB:

```
d = TLS.WavelengthStart;
TLS.WavelengthStart = d;
```

Sets or gets the start wavelength in nm.

## property WavelengthStop

## Type-Library:

```
get: HRESULT WavelengthStop([out, retval] DOUBLE* pVal);
put: HRESULT WavelengthStop([in] DOUBLE Val);
```

## MATLAB:

```
d = TLS.WavelengthStop;
TLS.WavelengthStop = d;
```

Sets or gets the stop wavelength in nm.

## property SweepRates

## Type-Library:

```
get: HRESULT SweepRates([out, retval] SAFEARRAY(double)*
pVal);
```

## MATLAB:

```
DoubleArray = TLS.SweepRates;
```

Returns a double array containing the sweep rates in nm/s that the current TLS can sweep at.

## property SweepRate

## Type-Library:

```
get: HRESULT SweepRate([out, retval] DOUBLE* pVal);
put: HRESULT SweepRate([in] DOUBLE Val);
```

## MATLAB:

```
d = TLS.SweepRate;
```

```
TLS.SweepRate = d;
```

Sets or gets the sweep rate in nm/s. Sweep rate  $v$  in nm/s and step size  $s$  in pm are related by  $0.3 \times v \text{ [nm/s]} = s \text{ [pm]}$  (e.g. 20 nm/s yields 6 pm step size)

#### property PWMSensitivity

Type-Library:

```
get: HRESULT PWMSensitivity([out, retval] DOUBLE* pVal);
```

```
put: HRESULT PWMSensitivity([in] DOUBLE Val);
```

MATLAB:

```
d = TLS.PWMSensitivity;
```

```
TLS.PWMSensitivity = d;
```

Sets or gets the power /current meter sensitivity for the DUT sweep in dBm. If Engine.PWMAutoranging is set to 1, this property will be adjusted automatically. This property has to be set in multiples of 10dBm (-20dBm to +10dBm).

#### property PWMSensitivityRef

Type-Library:

```
get: HRESULT PWMSensitivityRef([out, retval] DOUBLE* pVal);
```

```
put: HRESULT PWMSensitivityRef([in] DOUBLE Val);
```

MATLAB:

```
d = TLS.PWMSensitivityRef;
```

```
TLS.PWMSensitivityRef = d;
```

Sets or gets the power /current meter sensitivity for the reference sweep in dBm. If Engine.PWMAutoranging is set to 1, this property will be adjusted automatically. This property has to be set in multiples of 10dBm (-20dBm to +10dBm).

#### property RLMSensitivity

Type-Library:

```
get: HRESULT RLMSensitivity([out, retval] DOUBLE* pVal);
```

```
put: HRESULT RLMSensitivity([in] DOUBLE Val);
```

MATLAB:

```
d = TLS.RLMSensitivity;
TLS.RLMSensitivity = d;
```

Sets or gets the sensitivity of the return loss module (return path) in dBm. If Engine.PWMAutoranging is set to 1, this property will be adjusted automatically. This property has to be set in multiples of 10dBm (-60dBm to +10dBm).

Requires return loss module to be part of the optical setup.

property RLMSensitivityMon

Type-Library:

```
get: HRESULT RLMSensitivityMon([out, retval] DOUBLE* pVal);
put: HRESULT RLMSensitivityMon([in] DOUBLE Val);
```

MATLAB:

```
d = TLS.RLMSensitivityMon;
TLS.RLMSensitivityMon = d;
```

Sets or gets the sensitivity of the return loss module (monitor path) in dBm. If Engine.PWMAutoranging is set to 1, this property will be adjusted automatically. This property has to be set in multiples of 10dBm (-60dBm to +10dBm).

Requires a return loss module to be part of the optical setup.

property OutputPorts

Type-Library:

```
get: HRESULT OutputPorts([out, retval] SAFEARRAY(BSTR)* pVal);
```

MATLAB:

```
StringArray = MeasurementResult.OutputPorts;
```

Returns a string array containing the names of the possible output configurations of the current TLS. Will return "default" if only one output is supported by the current TLS.

property TLSOutputPort

Type-Library:

```
get: HRESULT TLSOutputPort([out, retval] LONG* pVal);
```

```
put: HRESULT TLSOutputPort([in] LONG Val);
```

MATLAB:

```
i = TLS.TLSOutputPort;
TLS.TLSOutputPort = i;
```

Sets or gets the output port to be used for the current TLS. This integer value refers to the available Output port configuration list obtained with `TLS.OutputPorts`.

#### property TLSPower

Type-Library:

```
get: HRESULT TLSPower([out, retval] DOUBLE* pVal);
put: HRESULT TLSPower([in] DOUBLE Val);
```

MATLAB:

```
d = TLS.TLSPower;
TLS.TLSPower = d;
```

Sets or gets the laser power in mW.

#### property OutputActive

Type-Library:

```
get: HRESULT OutputActive([out, retval] BOOL* pVal);
put: HRESULT OutputActive([in] BOOL Val);
```

MATLAB:

```
b = TLS.OutputActive;
TLS.OutputActive = b;
```

Enable/Disable the optical output for the current TLS. Setting this property will immediately turn on or off the optical output, while the other settings will only take effect upon the next sweep operation, e.g. running a single measurement.

#### property UseLambdaLog

Type-Library:

```
get: HRESULT UseLambdaLog([out, retval] BOOL* pVal);
put: HRESULT UseLambdaLog([in] BOOL Val);
```

MATLAB:

```
b = TLS.UseLambdaLog;
TLS.UseLambdaLog = b;
```

Enable/Disable wavelength logging (LambdaLog).

method LoadReference

Type-Library:

```
HRESULT LoadReference([in] BSTR Filename);
```

MATLAB:

```
TLS.LoadReference('Reference.pbin');
```

Loads a reference file in pbin format for the current TLS.

Note that the file must be present on the computer running the server, not the client. Use `Engine.ReferenceList` to get a list of reference files present on the server.

method SaveReference

Type-Library:

```
HRESULT SaveReference([in] BSTR Filename, [in] BOOL
Overwrite);
```

MATLAB:

```
TLS.SaveReference('Reference.pbin', 1);
```

Saves the current reference of the current TLS to a pbin file. Will not overwrite an existing file unless `Overwrite` parameter is set to 1.

Note that the file will be stored on the computer running the server, not the client.

property ReferenceDescription

Type-Library:

```
get: HRESULT ReferenceDescription([out, retval] BSTR* pVal);
```

MATLAB:

```
s = TLS.ReferenceDescription;
```

Gets a string containing the most important parameters of the current reference for the current TLS.

method ReferenceAvailableAtChannel

Type-Library:

```
HRESULT ReferenceAvailableAtChannel([in] LONG
nPort, [out, retval] BOOL* pVal);
```

MATLAB:

```
b = TLS.ReferenceAvailableAtChannel(nPort);
```

Use this method to check whether reference data for a specific port is available. Provide the port number to be checked, starting at 0 for the first port of the first power / current meter and counting continuously across multiple power / current meters (similar to the virtual ports described in [Performing a Reference Measurement](#) on page 51, except for port numbers starting at 0 here instead).

Returns TRUE if reference data is available or FALSE otherwise.

method ReferenceAveragePowerAtChannel

Type-Library:

```
HRESULT ReferenceAveragePowerAtChannel([in] LONG
nPort, [out, retval] DOUBLE* pVal);
```

MATLAB:

```
d = TLS.ReferenceAveragePowerAtChannel(nPort);
```

Use this method to get the average power of a reference trace for a specific port. Provide the port number to be checked, starting at 0 for the first port of the first power / current meter and counting continuously across multiple power / current meters (similar to the virtual ports described in [Performing a Reference Measurement](#) on page 51, except for port numbers starting at 0 here instead).

The average power is computed from the first, the last and the center sample of each trace. Will return NaN if no reference trace exists for the queried port.

property ERWavelengthStart

Type-Library:

```
get: HRESULT ERWavelengthStart([out, retval] DOUBLE* pVal);
put: HRESULT ERWavelengthStart([in] DOUBLE Val);
```

MATLAB:

```
d = TLS.ERWavelengthStart;
```

```
TLS.ERWavelengthStart = d;
```

Sets or gets the start wavelength in nm for use in Polarization Extinction Ratio measurements.

property ERWavelengthStop

Type-Library:

```
get: HRESULT ERWavelengthStop([out, retval] DOUBLE* pVal);
```

```
put: HRESULT ERWavelengthStop([in] DOUBLE Val);
```

MATLAB:

```
d = TLS.ERWavelengthStop;
```

```
TLS.ERWavelengthStop = d;
```

Sets or gets the stop wavelength in nm for use in Polarization Extinction Ratio measurements.

property ERUseSweepRange

Type-Library:

```
get: HRESULT ERUseSweepRange([out, retval] BOOL* pVal);
```

```
put: HRESULT ERUseSweepRange([in] BOOL Val);
```

MATLAB:

```
b = TLS.ERUseSweepRange;
```

```
TLS.ERUseSweepRange = b;
```

If set to true, Polarization Extinction Ratio measurements will use Engine.WavelengthStart and Engine.WavelengthStop, i.e., the parameters of the sweep measurement instead of the corresponding Engine.ERWavelengthStart / Stop parameters.

property ERWavelengthSamples

Type-Library:

```
get: HRESULT ERWavelengthSamples([out, retval] LONG* pVal);
```

```
put: HRESULT ERWavelengthSamples([in] LONG Val);
```

MATLAB:

```
i = TLS.ERWavelengthSamples;
```

```
TLS.ERWavelengthSamples = i;
```

Sets or gets the number of samples to be obtained during Polarization Extinction Ratio measurements. If set to 3, one sample will be obtained at Engine.ERWavelengthStart, one sample at Engine.ERWavelengthStop and one at the center wavelength. Otherwise the samples will be distributed accordingly.

#### property ERTLSPower

Type-Library:

```
get: HRESULT ERTLSPower([out, retval] DOUBLE* pVal);
```

```
put: HRESULT ERTLSPower([in] DOUBLE Val);
```

MATLAB:

```
d = TLS.ERTLSPower;
```

```
TLS.ERTLSPower = d;
```

Sets or gets the laser power in mW for use in Polarization Extinction Ratio measurements.

#### property ERPWMSensitivity

Type-Library:

```
get: HRESULT ERPWMSensitivity([out, retval] DOUBLE* pVal);
```

```
put: HRESULT ERPWMSensitivity([in] DOUBLE Val);
```

MATLAB:

```
d = TLS.ERPWMSensitivity;
```

```
TLS.ERPWMSensitivity = d;
```

Sets or gets the power / current meter sensitivity for the Polarization Extinction Ratio measurement in dBm.

If Engine.PWMAutoranging is set to true, this property will be adjusted automatically. This property has to be set in multiples of 10dBm (-20dBm to +10dBm).

If Engine.HighCapacitanceMode is set to true or a B29xxA instrument is used in the current configuration, this value is not allowed to be less than 0dBm.

## Alphabetical Automation Index

**A**

Activate, [104](#)  
Active, [105](#)

**B**

Bands, [135](#)  
BiasMode, [119](#)  
Busy, [105](#)

**C**

CalibrateRLM, [120](#)  
ClearReferenceChannels, [107](#)  
ClearReferenceChannelsSinglePort, [107](#)  
Configuration, [106](#)  
CopyReferenceChannel, [108](#)  
CopyReferenceChannelSinglePort, [108](#)

**D**

DarkIncludeMeasurement, [110](#)  
DataLoggingTimeout, [124](#)  
DeActivate, [105](#)  
DeleteEngine, [103](#)  
DeleteReferenceFile, [109](#)

**E**

ElectricalInputsPresent, [117](#)  
EmulationMode, [106](#)  
EnableILDeembedding, [128](#)  
EngineIDs, [103](#)  
ERIncludeMeasurement, [110](#)  
ERPWMSensitivity, [143](#)  
ERTLSPower, [143](#)  
ERUseSweepRange, [142](#)

ERWavelengthSamples, [142](#)  
ERWavelengthStart, [141](#)  
ERWavelengthStop, [142](#)  
EventMeasurementFinished, [121](#)  
EventPropertiesChanged, [121](#)  
EventreferenceOperationFinished, [121](#)

**F**

Fast Multichannel IL / PDL Setup, [15](#)  
FileSave, [112](#)  
ForceFullN7786BInitialization, [126](#)  
ForceMediumN7786BInitialization, [125](#)  
ForceTLSInitialization, [126](#)

**G**

GenerateDiodeCurrentData, [117](#)  
GenerateResponsivityData, [117](#)  
GenerateReturnLossData, [116](#)  
GenerateTETMData, [116](#)  
GetBias, [118](#)  
GetILDeembeddingFileList, [127](#)  
GetProtocolTextAt, [122](#)  
GetPWMChannelCount, [114](#)  
GetTLSInterface, [114](#)

**H**

HighCapacitanceMode, [110](#)

**I**

ILDeembeddingTargetFolder, [128](#)  
ImmediateEvaluation, [124](#)  
Include, [135](#)  
Interface “IEngine”, [104](#)

Interface “IEngineMgr”, [102](#)  
Interface “IEngineTLS”, [135](#)  
IsBiasEnabled, [118](#)  
IsVersionGreaterOrEqual, [102, 104](#)

**K**

KeepRawData, [115](#)

**L**

LambdaZeroingMode, [119](#)  
LoadConfiguration, [106](#)  
LoadReference, [140](#)

**M**

MaxAllowedRefLoss, [113](#)  
MeasurementResult, [112](#)  
MeasurementSpeedLegacyMode, [125](#)  
ModelCode, [135](#)  
MPPMAutoGain, [115](#)

**N**

NewEngine, [102](#)

**O**

OpenEngine, [103](#)  
OutputActive, [139](#)  
OutputPorts, [138](#)

**P**

property  
  StaticPolarizationMode, [130](#)  
ProtocolMax, [122](#)

- ProtocolMin, [122](#)
  - ProtocolText, [121](#)
  - PWMAutoRanging, [115](#)
  - PWMCount, [113](#)
  - PWMSensitivity, [137](#)
  - PWMSensitivityRef, [137](#)
- Q**
- QuickPolarimeterSetup, [124](#)
- R**
- ReadCurrentStabilizerState, [134](#)
  - ReadCurrentStokesParameters, [134](#)
  - ReadCurrentWaveplateOrientations, [134](#)
  - ReCalculate, [113](#)
  - ReferenceAvailableAtChannel, [141](#)
  - ReferenceAveragePowerAtChannel, [141](#)
  - ReferenceDescription, [140](#)
  - ReferenceList, [109](#)
  - Resolution, [113](#)
  - ResolutionValues, [112](#)
  - ReturnLossModulePresent, [117](#)
  - RLMCalReflectorValue, [120](#)
  - RLMSensitivity, [137](#)
  - RLMSensitivityMon, [138](#)
- S**
- SaveReference, [140](#)
  - SetBias, [118](#)
  - SetILDeembeddingFileList, [127](#)
  - SetStatic, [129](#)
  - SOPSearchQuickMode, [125](#)
  - StartCustomReference, [107](#)
  - StartCustomReferenceSinglePort, [107](#)
  - StartILDeembeddingMasterPortMeasurement, [126](#)
  - StartILDeembeddingSpecificPathMeasurement, [126](#)
  - StartMeasurement, [111](#)
  - StartMeasurementRepeat, [111](#)
  - StartReference, [106](#)
  - StaticFilename, [132](#)
  - StaticPolarizationMaxMinMode, [131](#)
  - StaticPolarizationMode, [130](#)
  - StaticPolarizationPSPMode, [132](#)
  - StaticPWMPort, [130](#)
  - StaticStokesParameters, [133](#)
  - StaticTLSPower, [130](#)
  - StaticWavelength, [129](#)
  - StaticWaveplateOrientations, [133](#)
  - StopMeasurement, [111](#)
  - StopStabilizer, [129](#)
  - SweepRate, [136](#)
  - SweepRates, [136](#)
- T**
- TLSCount, [114](#)
  - TLSExtendedSweepRange, [116](#)
  - TLSOutputPort, [138](#)
  - TLSPower, [139](#)
- U**
- UndoReferenceOperation, [108](#)
  - UseLambdaLog, [139](#)
  - UserInputChoice, [123](#)
  - UserInputPrompt, [123](#)
  - UserInputResponse, [123](#)
  - UserInputWaiting, [122](#)
- V**
- ValidateSettings, [109](#)
  - ValidateSettingsNoRefCheck, [109](#)
  - Version, [102, 104](#)
- W**
- WavelengthStart, [136](#)
  - WavelengthStop, [136](#)
- Z**
- ZeroAllMPPMs, [119](#)
  - ZeroAllTLS, [120](#)



# 5 Troubleshooting

Symptoms and Solutions / 148

## Symptoms and Solutions

**No measurement data shown**

Possible Reason	Solution
An error occurred during the measurement, preventing the software from fully evaluating the data.	Check the output in the measurement status box for errors.
The trace contains invalid data only.	Under certain circumstances (e.g. certain over-range conditions, or bad/outdated power / current meter or TLS zeroing data, certain regions may contain NaN values instead of reasonable data. NaN values are not displayed in the engine GUIs / File Viewer.
There is no reference data available for the specific port(s).	Check Port/Reference Manager.
The specific graph type is not shown in the graph area of the engine GUI / File Viewer.	Especially when switching between different PAS engines or using the File Viewer for viewing data from different PAS engines, often different graph types are selected to be shown in the graph area. This is done by expanding the measurement in the browser tree, right-clicking the corresponding data type (e.g. Insertion Loss or PDL), then selecting <b>View Graph</b> . Usually unused graphs are removed from the graph area then, by right-clicking into one of the graphs and selecting <b>Remove Graph</b> . This view configuration is shared by the File Viewer and the engine GUIs, so it can happen that the graph types used by the current engine have been removed from the graph area from within another engine. See the N7700 User's Guide for further details.

### Measurement Status Box not visible

Possible Reason	Solution
Window minimized.	If you minimized the status window, it will stay that way until a user input is required. You can get it back to front anytime by clicking its entry in the task bar.
Window in background.	Unless "always on top" is selected, the measurement status box may get behind other windows. You can get it back to front by clicking its entry in the task bar.
Window closed.	If you closed the status window, it will stay that way until a user input is required. You can get it back by selecting <b>Measurement Status</b> in the <b>View</b> menu.
Window position stored in screen region that is currently invisible.	This could happen when working with multiple screens or varying screen resolutions. By selecting <b>Reset Status Window Position</b> in the <b>View</b> menu, the measurement status box position will be reset to the current top left corner of the engine window.

### Connection Expert displaying a warning icon next to an N778xB instrument connected to GPIB

Possible Reason	Solution
N778xB instruments return an IDN string that differs slightly from most other Keysight instruments, resulting in a warning icon next to the instrument.	You can safely ignore this warning.

### Connection Expert not showing an N778xB instrument connected to USB

Possible Reason	Solution
By design N778xB instruments are not listed in the Connection Expert.	Use the Polarization Navigator or tools from the Photonic Application Suite for checking and addressing N778xB instruments.

### Can't find any instrument connected to GPIB

Possible Reason	Solution
If you're using a GPIB interface connected to your PC and an N778xB instrument is connected to the GPIB bus and it is configured for Remote GPIB, communication on the GPIB bus doesn't work.	Power-cycle the N778xB instrument to set it back to default settings (Remote GPIB off). Refer to <a href="#">Using the N7786B GPIB Interface (Remote GPIB)</a> on page 31 for further details.

### Can't find N778xB instrument connected to USB

Possible Reason	Solution
When running a version of the Photonic Application Suite below 1.5 on 64-bit operating systems, there is no USB support for N778x instruments.	Upgrade to Photonic Application Suite 1.5 or higher.

### The list of values for a drop-down control appears invisible or displaced

Possible Reason	Solution
Different scaling factors for the different screens in some multi-screen setups.	Ensure that the scaling factor for all the screens is consistent.
	Consider using a single screen setup, define another screen as the default screen, or start the engine from an Explorer window on the screen that the engine GUI is intended to show on.
The engine window is maximized.	To make the invisible list of values appear on the screen, restore the window to its normal size ensuring that it is not too close to the top left edge of the screen. Then set the required value in the drop-down control. Note that the list of values will still appear displaced.
	If the list of values is visible (although displaced), select the drop-down list and use the cursor (arrow) and Enter keys to navigate to and select the required value.

### COM API registration fails

Possible Reason	Solution
MATLAB Compiler Runtime (MCR) has not been installed prior to measurement engine installation.	Register the COM API after later MCR installation or re-register for any other reasons, by running the <b>Re-register all N7700 COM APIs</b> program from the Start menu.

#### NOTE

A troubleshooting utility called **Get N7700 System Information** has been introduced to help Keysight technical support gather some vital details required to resolve problems with the software at the user end, engine startup problems, for instance. This utility is accessible from the Start menu and should be run to capture information that can be shared with Keysight technical support.

