VPItransmissionMaker/VPIcomponentMaker

Release Notes (Excerpt)

How to contact VPIphotonics GmbH

www.VPIphotonics.com	Web site
info@VPIphotonics.com	General information
sales@VPIphotonics.com	Sales, ordering information
support@VPIphotonics.com	Technical support, license key requests

April 2016. Document Part Number: TC0-RN01-26 16109

VPItransmissionMaker/ VPIcomponentMaker Version 9.7 Release Notes

VPIphotonics is proud to announce the following new features and feature enhancements for VPItransmissionMaker/VPIcomponentMaker, which include the products:

- VPItransmissionMaker™Optical Systems Version 9.7
- VPIlabExpert™ Version 9.7
- VPIcomponentMaker™Fiber Optics Version 9.7
- VPIcomponentMaker™Photonic Circuits Version 9.7
- VPIplayer™ Version 9.7

This document only covers the *NEW* features included in this release. Details of the existing features can be found in the product manuals for Version 9.7.

This release will upgrade VPItransmissionMaker/VPIcomponentMaker Version 9.5 (released in May 2015).

For further updates, visit the *VPIphotonics Users Forum* at forums.vpiphotonics.com.

Disclaimer

VPIphotonics reserves the right to change the product specifications without notice.

System Requirements and Recommendations

Hardware Requirements

VPItransmissionMaker/VPIcomponentMaker requires the following minimum hardware configuration:

- Personal computer with 1 GHz or faster 64-bit (x64) processor
- 2GB RAM *(more is recommended for memory-intensive application scenarios)*
- 2 GB of free hard disk space for software installation process,
  500 MB will be used by VPItransmissionMaker after the installation,
  1 GB additionally recommended for file storage
- XGA monitor with 1024x768 pixels minimum display resolution (*higher resolutions recommended*)
- DirectX 9 graphics device with WDDM 1.0 or higher driver
- For GPU-assisted computations NVIDIA® video adapter with compute capability (version) greater than or equal to 1.3 and double-precision support is required
- DVD-ROM drive

Supported Platforms

VPItransmissionMaker/VPIcomponentMaker supports the following platforms:

- Windows® 7 Enterprise SP 1 *(64-bit version)*
- Windows® 8.1 Enterprise *(64-bit version)*
- Windows® 10 Enterprise *(64-bit version)*

Note: The software may also run on other (similar) operating system versions. Contact support@vpiphotonics.com for more information.

Software Requirements

VPItransmissionMaker/VPIcomponentMaker requires the following minimum software configuration:

- One of the supported platforms, as listed above
- TCP/IP protocol installed and configured
- Microsoft .NET Framework 4.0 *(installed automatically)*
- Python 2.7.3 *(64-bit version installed automatically, including NumPy 1.9.0)*
- A PDF file reader is required to display product documentation
- Latest NVIDIA® drivers if GPU-assisted computations are required

Cosimulation

VPItransmissionMaker/VPIcomponentMaker supports cosimulation with third party software. Among others, interoperability with the following software is supported:

- The MathWorks MATLAB® version 8.6 R2015b *(64-bit version)*
- Python 2.7.3 *(64-bit version, including NumPy 1.9.0 and SciPy 0.14.0)*
- Keysight Technologies Advanced Design System (ADS) 2015.01

What's New in Version 9.7

VPIphotonics Design Suite, aka VPItransmissionMaker™/VPIcomponentMaker™, Version 9.7 provides common usability, design process and data analysis capabilities while offering access to the following application-specific products:

- VPItransmissionMaker™Optical Systems Version 9.7
- VPIlabExpert™ Version 9.7
- VPIcomponentMaker™Fiber Optics Version 9.7
- VPIcomponentMaker™Photonic Circuits Version 9.7
- VPIplayer™ Version 9.7

Among the new features compared to the previous versions are enhancements of the user interface, in particular a better support for creation of complex parameter expressions, as well as advances in simulation capabilities: support of multicore fiber systems, N-dimensional modulations formats, new lab equipment, etc.

VPIphotonics' software solutions have proven to be beneficial in winning and successfully performing many research and design projects — for commercial companies as well as for educational institutions. With the improved capabilities provided in Version 9.7, our modeling suite is set to deliver the same outstanding results in the future.

Supported Platforms

Starting with version 9.7, VPIphotonics Design Suite is available as a 64-bit version running on 64-bit operating systems only. See more information on supported platforms in "Supported Platforms" on page 2.

Licensing

VPItransmissionMaker/VPIcomponentMaker 9.7 comes with a new version of the VPIlicenseServer that will be automatically installed together with the VPItransmissionMaker/VPIcomponentMaker software. In configurations with a remote license server, the latest version of the license server should be installed on the server computer. Note that VPItransmissionMaker/VPIcomponentMaker 9.7 will not work with older versions of VPIlicenseServer (in this case you may receive the "Version of vendor daemon is too old" error message). At the same time, older versions of VPItransmissionMaker/VPIcomponentMaker will work with the new version of the license server; however, an update of the license file will be required. In this case, please contact us for support on requesting a new license.

Photonic Design Environment

The Photonic Design Environment (PDE) of Version 9.7 represents a powerful user interface offering a high level of usability and personalization.

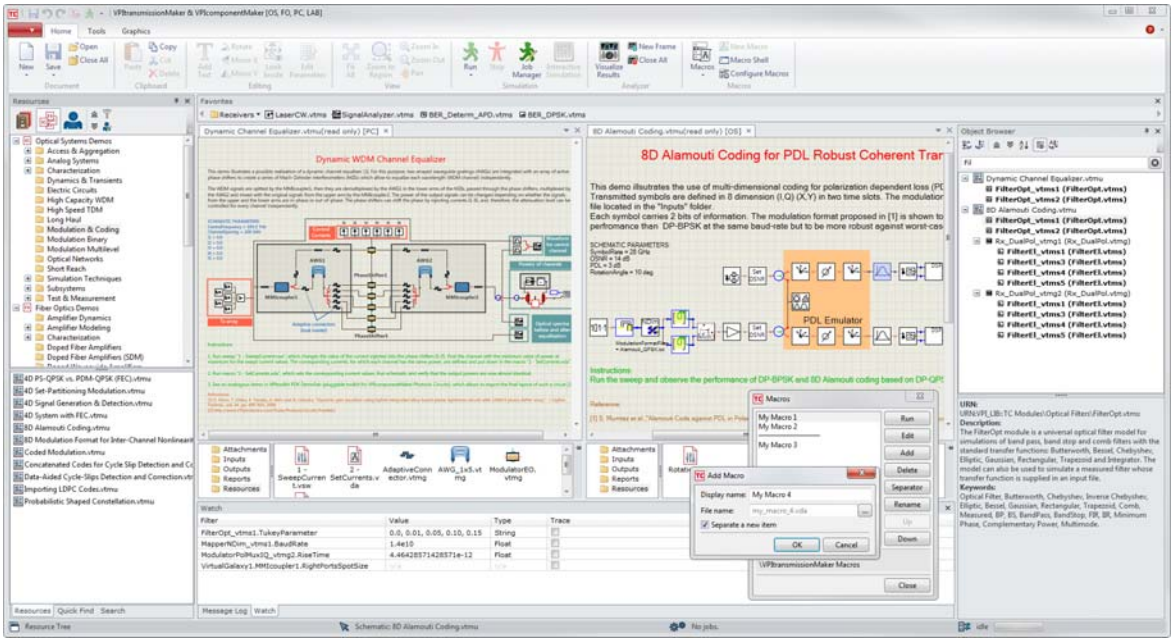


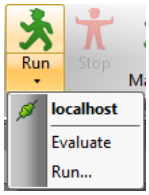
Figure 12 *Photonic Design Environment (PDE) of Version 9.7*

For detailed reference information and instructions on using the features of the Photonic Design Environment, see the *VPItransmissionMaker™/ VPIcomponentMaker™ User Interface Reference* and *VPItransmissionMaker™/ VPIcomponentMaker™ Simulation Guide*.

Parameter Watch — Easy Evaluation of Parameter Values

Creating sophisticated simulations has been made significantly easier with the introduction of the new Parameter Watch functionality. It allows you to display the actual value assigned to a parameter directly in the PDE. This is especially useful for the creation of complex Tcl or Python expressions or initialization scripts.

Parameters to observe may be specified directly from the **Parameter Editor** via the drop-down menu of the parameter in question. More generic filters (also called *loose* filters), which enable you to trace parameters with similar names but for different module instances, may be added directly in the **Watch** panel.



A special evaluation mode of simulation setups has been introduced to perform the evaluation of parameter values without the necessity to actually execute the (possibly long lasting) simulation.

The parameter values may be observed using the **Watch** panel or traced via the **Message Log**. Entries in the **Message Log** contain a link that can be used for quick navigation to the module whose parameter value is traced.

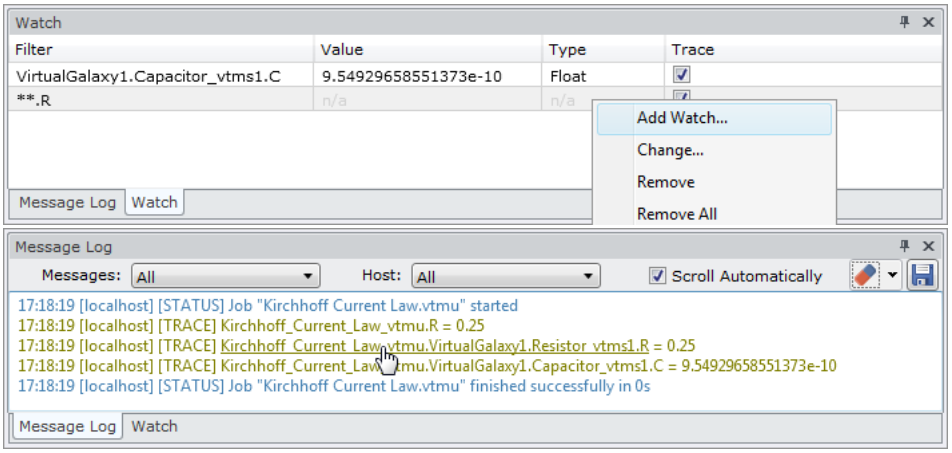
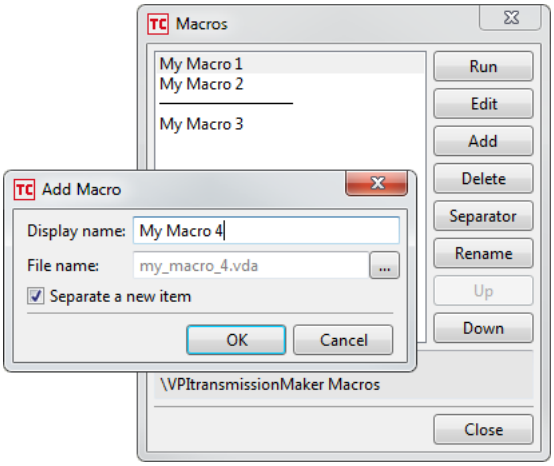


Figure 13 Watch panel (top), tracing parameter values in the Message Log (bottom)

Macros Manager

User macros available through the **Macros** button menu on the **Home** tab of the ribbon can now easily be managed via the **Macros Manager** dialog available upon pressing the **Configure Macros** button on the same tab.



It is possible to add a copy of the existing macro or create a new one from the template, edit a particular macro or reorganize the macros sequence in the **Macros** menu.

The location of user macros has been changed. They are now stored in the user's documents folder (e.g., `c:\Users\user1\Documents\VPItransmissionMaker Macros`).

More Enhancements

Other important new features and improvements include:

- When parameter values are read from a `*.vtmp` file and there is no corresponding parameter in the target schematic, such parameters are now created automatically. Parameters that will be created are explicitly marked in the **Load Parameter Settings** dialog.
- Instance IDs of modules that produced messages are represented with links in the **Message Log**. When this link is clicked, the corresponding module is selected on the schematic and its **Parameter Editor** is opened.
- Warning and information messages are distinguished by color in the **Message Log** window. Overall, the message format has been made cleaner.
- The new macro command `writemessage` has been introduced to allow macros to write directly to the **Message Log**.
- The PDE context menu has been made more responsive by requesting the VPIphotonicsAnalyzer menu in the background.

Module Developments

Multimode / Multicore Transmission

The multimode signal model has been extended to support multicore fibers (s. example in Figure 14), including heterogeneous multicore fibers with cores that can have different parameters and support different number of modes. The parameters of multicore fibers can be specified via the ***SolverFiberMeasuredMM*** module. Modules that supported multimode signals in the previous version already (e.g., fibers, couplers, amplifiers, etc.) now support signals of multicore fibers as well.

The signal propagation in multicore fibers can be simulated using the ***FiberMM*** module. While mode coupling within each core can be accounted for, distributed coupling between the modes of different fiber cores is ignored in this version. Furthermore, discrete coupling between modes can be simulated using the ***CouplerFiberMM*** module. Signal amplification can be simulated with the system-level ***AmpSysOptMM*** module that can take into account core/mode-dependent gain and noise characteristics.

Other improvements supporting the simulation of space division multiplexing (SDM) systems include:

- The modules ***SolverFiberMM*** and ***SolverDopedFiberMM*** allow the user to limit the range of simulated modes. If this limiting is active, only the modes that are specified by the module parameters will be considered in signal propagation and coupling. This allows you to reduce complexity of the multimode simulations by ignoring unimportant fiber modes.
- Mode-dependent attenuation (background loss) is now considered in both ***SolverFiberMM*** and ***SolverDopedFiberMM*** modules. Its characteristic can be specified either by parameters or an input file (for wavelength-dependent loss).
- The spectral dependence of the fiber index profile can be accounted for more accurately in the ***SolverDopedFiberMM*** module. While in previous versions the wavelength dependence of the refractive index was assumed to be independent of the fiber radius (which is a fair approximation for many applications), the

transversal index distribution can now be specified independently for different wavelengths as well.

- The modules ***CombinerSplitterMM*** and ***CouplerFiberMM*** support now polarization-dependent coupling using the Jones matrix description.
- The ***DopedFiberMM*** module has been enhanced to simulate mode coupling using the coupled power equations. The coupling strength between modes can be either specified directly by the power coupling coefficients, or calculated from the field coupling coefficients, coupling correlation length and correlation function.

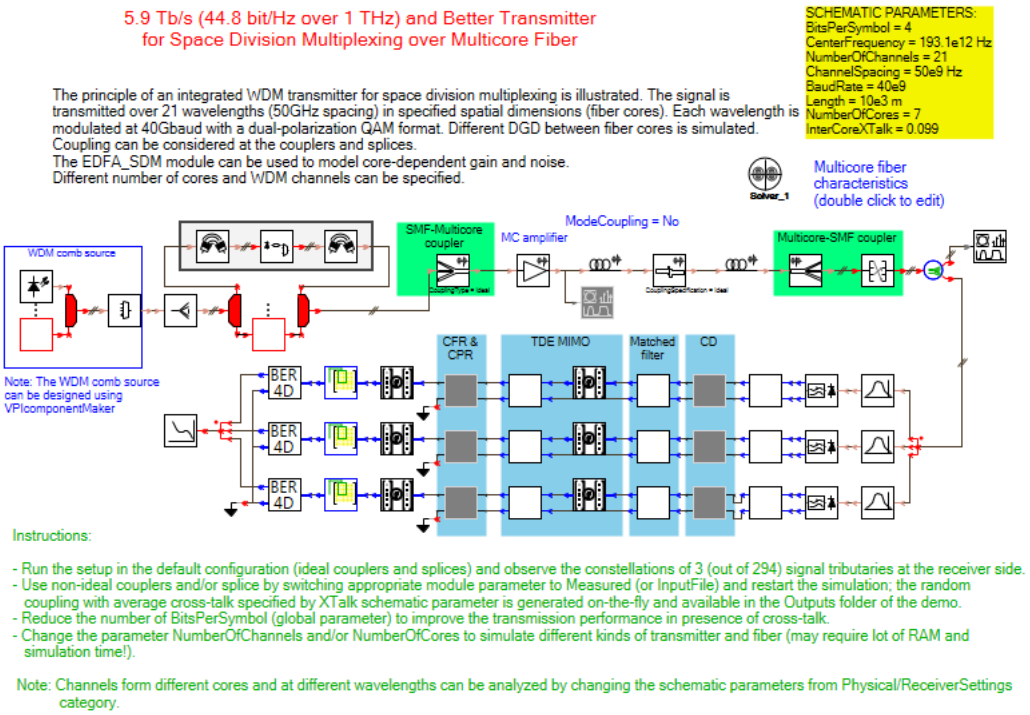


Figure 14 *High-capacity SDM-WDM transmission over multicore fiber. The number of WDM channels, fiber cores and channels that are analyzed at the receiver can be easily specified by the simulation parameters.*

New and Enhanced DSP Algorithms

In order to better support the generation of a superchannel using higher order modulation formats (32, 64 & 128QAM), the following algorithms have been added/improved in the ***DSP_Lib*** module:

- Digital Nyquist filtering, which can be used, for instance, to perform realistic pulse shaping
- Blind Phase Search (BPS) algorithm for carrier phase recovery (CPR)
- Generic multi-modulus algorithm (MMA) for optimal equalization of higher-order modulation using adaptive time-domain MIMO filter.

Furthermore, the Digital Back Propagation (DBP) algorithm has been updated to support log-spaced step sizes.

N-Dimensional Modulation

Extending the symbol coding beyond the classical four dimensions of the optical field (I and Q in two orthogonal polarizations) has been shown to increase the power efficiency and noise sensitivity, but also the nonlinear tolerance of the transmitted signal. Additional dimensions include time (e.g., time-hybrid modulation) where each slot represents a new dimension, or space — in SDM systems, where each mode represents a spatial dimension.

The new ***MapperNdim*** and ***BER_Ndim*** modules support the generation, detection and analysis of an arbitrary N-dimensional modulation format. A set of available dimensions and the applied modulation format (defined by its bit-to-symbol mapping) are specified in a text file. An excerpt from the modulation format file is shown below, where all possible coding dimensions are utilized:

```
#    Slots_2
#    Dimensions_2
#    X, Y
#    I, Q
// all possible dimensions are used in coding:
// two non-time dimensions, dim1 and dim2, two time slots
// X and Y polarizations and I and Q components
//                            slot1          |        slot2
//                     dim1         dim2     |   dim1        dim2
//                   X     Y      X     Y    |  X     Y     X     Y
//                  I Q  I  Q    I Q  I  Q   | I Q  I  Q   I Q  I  Q
0000000000000000  -3 3 -3  3   -3 3 -3  3    -3 3 -3  3  -3 3 -3  3
0000000000000001  -3 3 -3 -1   -3 3 -3 -1    -3 3 -3 -1  -3 3 -3 -1
0000000000000010  -3 3  1  3   -3 3  1  3    -3 3  1  3  -3 3  1  3
0000000000000011  -3 3  1 -1   -3 3  1 -1    -3 3  1 -1  -3 3  1 -1

...
```

The ***BER_Ndim*** module provides the same capability as the ***BER_4D*** module: BER and SER estimation using error counting or Gaussian approximation, LLR computation, symbol-to-bits decoding, and automatic constellation alignment. Figure 15 shows a simulation setup and results for a comparison of DP-BPSK and 8D-modulation coding formats.

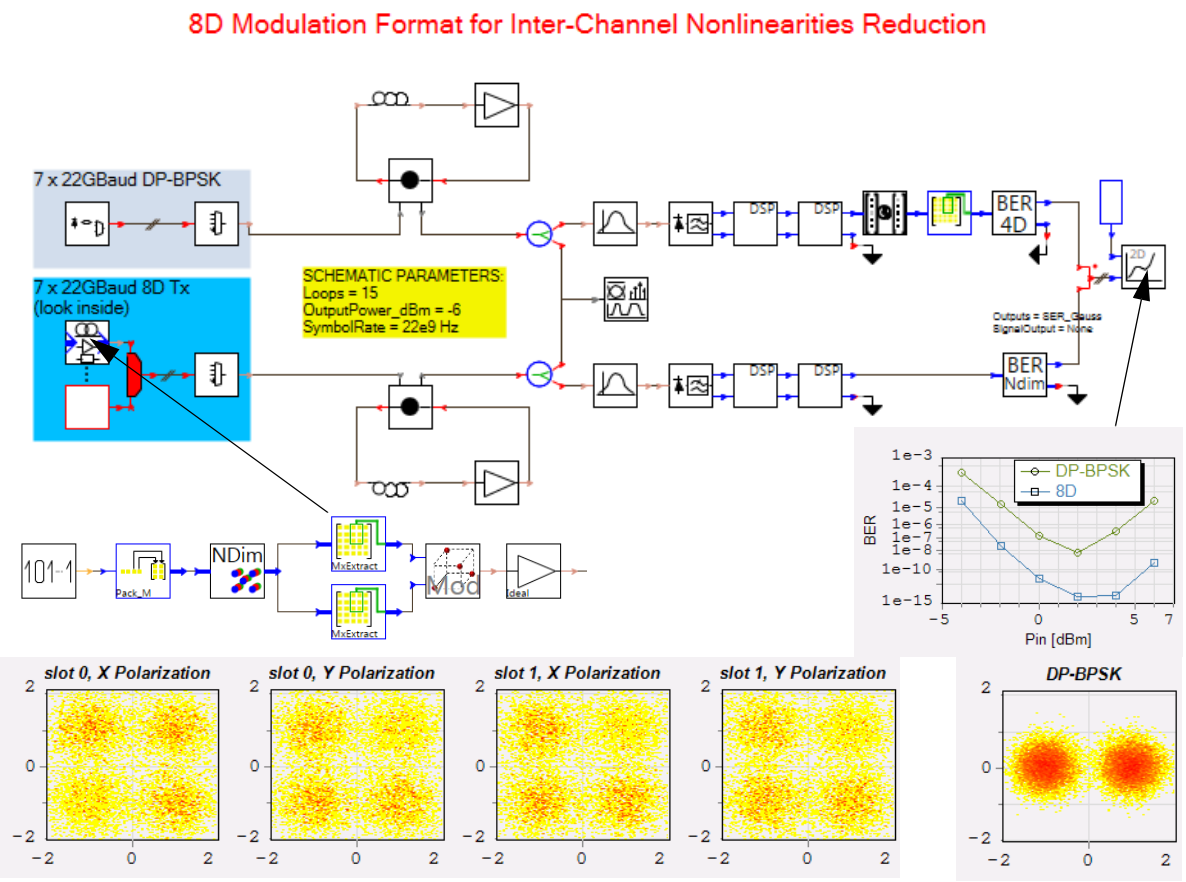


Figure 15 Example of n-dimensional modulation

PAM - Estimation of Q and EO

Besides bit- and symbol-error-rate (BER, SER) it is now possible to estimate the Q-factor and eye opening (EO) of amplitude modulated multilevel signals including pulse amplitude modulation (PAM).

Adaptive Equalizer

The existing functionality of feed-forward equalizer (FFE), decision feedback equalizer (DFE) and nonlinear Volterra equalizer has been merged and made available via the new module ***AdaptiveEqualizer***. This simplifies comparative studies of different equalizer types and settings within a single simulation setup.

Lab Equipment

The new VPIlabExpert modules ***Anritsu_MS2830A_VSG*** and ***Anritsu_MS2830A_Scope*** provide a seamless integration of the Anritsu MS2830A Vector Signal Generator (VSG) and Signal Analyzer in the simulation environment enabling a better support of radio-over-fiber, analog photonics or wireless applications. Other Anritsu devices from the same product family are supported as well.

PhotonicsTLM

The usability of the ***PhotonicsTLM*** module has been enhanced by adding the following two features:

- The amplified spontaneous emission (ASE) spectrum is now decoupled by default from the noise generation spectrum. For this, the parameter **`NoiseBandwidth`** has been extended to support the generation of flat spectral noise with de facto infinitely large **`NoiseBandwidth`** using its new default value of 0. This allows that the noise is generated much faster (in case of applying a parabolic or measured gain shape model) and the resulting ASE spectrum stays in full agreement with any used gain spectrum model. But setting **`NoiseBandwidth`** to any nonzero value allows you to restore the old ***PhotonicsTLM*** functionality and to additionally adjust the noise generation spectrum according to any user-defined Lorentzian shape. See *VPIcomponentMaker™Photonic Circuits User's Manual*, section *"Spontaneous Emission Filtering"* and the demonstration "Photonic Circuits Demos > Getting Started > PhotonicsTLM - Spontaneous Emission Models" for details.
- The width of the bus connected to the electrical output port CarrierDensityOutput may now be different from the value (2***`NumberOfDeviceSections`**) that has been expected before. If the bus width is smaller than expected, all extra output signals are assumed to be sent to ***Ground***. If the bus width is larger than expected, all missing signals are assumed to be fed by ***NullSource***. This simplifies the usage of the ***PhotonicsTLM*** module with varying number of device sections, especially in the frequently encountered case when the output signals from the CarrierDensityOutput port are merely ignored and sent to ***Ground***. The order of output signals does not depend on the bus width and is described in the *VPIcomponentMaker™Photonic Circuits User's Manual* under *"Output of Carrier Density, Current and Voltage"*.

Scripting and Cosimulation

A number of improvements in version 9.7 enhance the scripting and cosimulation capabilities of VPItransmissionMaker™/VPIcomponentMaker™:

- Function parameters have been made available for cosimulation modules. Such parameters can be used to describe functional dependencies instead of constant values specified by regular parameters. Examples include support of arbitrary user-defined expressions and measured data files for frequency-dependent effective mode index and attenuation, voltage-dependent phase shift and electroabsorption, carrier-dependent gain, position-dependent apodization and

chirp profiles in gratings, and so forth. In contrast to regular parameters, function parameters in cosimulation are represented by complex objects which allow you to distinguish the parameter expression and expression type. For most cosimulation environments (except MATLAB) and for initialization scripts ("`init.py`" files), it is also possible to directly evaluate function parameters defined by Python expressions in the context of the appropriate upper-level initialization script.

- File and directory parameters are now handled more intelligently so that there is no need to expand them directly in the code of the cosimulation modules and initialization scrips anymore — absolute paths are sent directly.
- New functions `get_context_id()` and `get_inputs_dir()` are made available for Python cosimulation and Python initialization scripts. The first one returns the full topology path to the module instance (e.g., `U1_vtmu.G1_vtmg1` or `U1_vtmu.G2_vtmg1.CoSimInterface_vtms1`). The second one is intended to return the full path to the `Inputs` folder of a galaxy or universe that contains an instance of the ***CoSimInterface*** module (or initialization script). This simplifies creation of custom modules with complex functionality such as support of an arbitrary user-defined temperature in active optoelectronic devices on the basis of carrier-dependent gain spectra measured for only several temperature points, support of elastic connectors with length values being resolved in background by calling external layout design tools, and the like.
- Automatic value conversion has been improved for Python expressions. For example, it is now possible to use native Python lists and tuples to define the value of an array parameter.
- The Library cosimulation now uses 64-bit integers for representation of quantized signal properties (e.g., signal frequencies) to avoid integer overflow, which could occur in previous versions (using 32-bits integers) for long `TimeWindow` values. Although the cosimulation code compiled for the previous versions will work with Version 9.7, we strongly recommend to recompile it in order to overcome the aforementioned limitation.
- The support for debugging Python scripts using third-party debuggers has been improved. After adding this feature, the encoding of Python cosimulation and initialization scripts should strictly follow the Python encoding rules (for more details, see the official Python tutorial https://docs.python.org/2.7/tutorial/interpreter.html#source-code-encoding).

VPIphotonicsAnalyzer

Visualization of Multimode Signals

The visualization and analysis support of multimode signals in VPIphotonicsAnalyzer has been further improved, including signals in multicore fibers. In addition to the mode IDs (mode numbers), signals can be selected by using information of core number, radial and azimuthal index. The mode solver referenced in the signal is displayed in the input selector. An exemplary Scope chart displaying a multimode signal is shown in Figure 16.

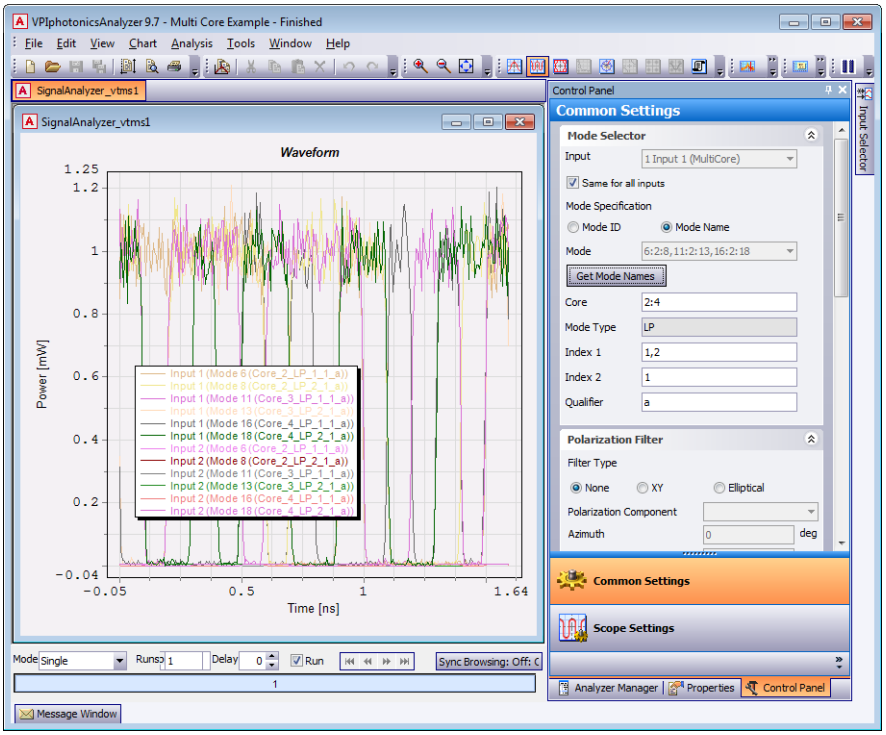


Figure 16 Visualization of multimode signals

Analysis of Signal Metrics

A number of metrics for electrical signals in time domain can be calculated now. The metrics are available from the **Scope Analysis** tab and include **Power**, **RMS**, **Peak Positive/Negative**, **CF** (Crest factor), and **PAPR** (peak-to-average power ratio). The metrics can be calculated for both, the whole signal and its AC and DC components.

Additional metrics can be calculated in the **Constellation** mode of the Analyzer: **Offset**, **Phase Error** and **Frequency Error** serve to estimate imperfections of the received constellation.

General Features

Furthermore, a number of general visualization and analysis features are provided:

- The chart properties **`Chart Title`** and **`Legend Text`** are built as templates now (like **`Axis Title`**). For example, **`Chart Title`** is set by default to `{$(ChartTitleSim)}`, where the value of **`ChartTitleSim`** is defined by the parameters of the module that produces the corresponding Analyzer window.

  For modules with internal visualizers, this approach allows you to simultaneously display parameter-dependent and user-specified information in the chart title and legend. For more details, see *VPItransmissionMaker™/ VPIcomponentMaker™ User Interface Reference* under "Internal Visualizers".
- Up to ten Point Markers can be added to the chart. The point markers can be used to "bookmark" features on a signal trace, for example, peaks of a signal spectrum.

  The behavior of a point marker for the subsequent simulation runs depends on the settings of the marker properties **`Stick to X`** and **`Stick to Y`**. For instance, if **`Stick to X`** is set to `Yes` while **`Stick to Y`** is set to `No`, the marker will keep its x-coordinate and change the y-coordinate so that the crossing point with the signal is not lost.

  If a pair of point markers is selected, the label of the second marker will show both the marker coordinates and their difference.
- The status of the simulation ("*Running...*" or "*Finished*") is shown in the Analyzer Frame title next to the schematic name.
- It is possible to limit the number of signals that are stored in the Analyzer after they are produced at different iterations/simulation runs. In many VPIlabExpert applications, data is continuously sent to the simulator from the experimental instrumentation in real time. Limiting the number of stored iterations will limit the disk space occupied by temporary Analyzer files.

Replacement of Obsolete Modules

With the previous versions, a number of other modules have also been superseded by newer, enhanced models. The old, obsolete modules are still available in the **Module Library** allowing continued access if needed. To show obsolete modules, select the corresponding option on the **General** tab of the application preferences. When this option is selected, obsolete modules appear in the tree alongside current modules with a line through the names. The *Replacement Wizard* simplifies the process of updating previously created schematics and custom modules. Even though some setups continue to work with obsolete modules, it is recommended to perform the module replacement step by step, as obsolete modules will be removed from the tree altogether in future product releases.

The module ***DebugInfoOutputTk*** has been made obsolete — its functionality is replaced by the ***SignalAnalyzer*** module in **SignalStructure** mode.

Applications Demonstrations

Existing demonstrations have been modified and new ones added to illustrate the functionality and applications of the new features and modules. There are over 800 demonstrations available in the VPIphotonics Design Suite now. Among the new and updated demonstrations in this release are the following:

Optical Systems

Access & Aggregation → Digital SCM
→ MultiCAP
Access & Aggregation → PON
→ 4PAM for 400G Ethernet over SMF
Characterization → Dispersion & Kerr
→ Inter-Modal FWM
High Capacity WDM
→ 12D Coding for SDM
→ Simulation of Multicore Fibers - Getting Started
→ Ultra High Capacity for SDM System (UPD)
Modulation & Coding
→ 8D Alamouti Coding
→ 8D Modulation Format for Inter-Channel Nonlinearities Reduction
→ Coded Modulation
→ Importing LDPC Codes
→ Probabilistic Shaped Constellation
Modulation Multilevel
→ DSP for 32QAM
Simulation Techniques → Cosimulation → Library
→ Function Parameters in Cosimulation (Lib)
Simulation Techniques → Cosimulation → Python
→ Function Parameters in Cosimulation (Python)
Simulation Techniques → General
→ Downsampling Aperiodic Optical Signals
→ Multiple Runs and DSP
Subsystems → Receiver Electronics
→ Volterra Equalizer (UPD)
Test & Measurement → Importing Data
→ Coupling Computation using Zemax Files

Fiber Optics

Doped Fiber Amplifiers (SDM)
→ DFAMM - Mode Coupling
Simulation Techniques → Cosimulation → Library

→ Function Parameters in Cosimulation (Lib)

Simulation Techniques → Cosimulation → Python

→ Function Parameters in Cosimulation (Python)

Simulation Techniques → General

→ Downsampling Aperiodic Optical Signals

Photonic Circuits

Getting Started

→ Measured Transfer Function

→ PhotonicsTLM - Spontaneous Emission Models

Laser Characterization

→ Intensity Noise in Multimode Lasers

Lasers & SOAs

→ Soliton with Mode Locked Laser

Optical Signal Processing

→ Dynamic Channel Equalizer

Simulation Techniques → Cosimulation → Library

→ Function Parameters in Cosimulation (Lib)

→ Optical Signal Generation (Lib)

Simulation Techniques → Cosimulation → Python

→ Function Parameters in Cosimulation (Python)

→ Optical Filter (Python)

→ Optical Power Meter (Python)

→ Photodiode (Python)

Simulation Techniques → General

→ Downsampling Aperiodic Optical Signals

Passive Circuits

→ Optimization of Ring-Loaded Unbalanced MZI (UPD)

Systems

→ AWG - Impact of Passband Shape for OOK

→ Cascaded AWG 100-Channel PIC

Transmitters & Receivers

→ Silicon Traveling-Wave Modulator

Ultrafast Devices

→ Soliton Pulse Compression

Lab Expert

Lab-Ready Setups→ Systems

→ RoF Link

→ PAM Transmission for High-Speed Short-Haul & Access Systems (UPD)

Modulation & Coding

→ Importing LDPC Codes

New application demonstrations covering current topics and novel simulation techniques are also available on the VPIphotonics *Users Forum* at forums.VPIphotonics.com.

Documentation

The accompanying documentation has been revised to provide direct access to reference material from the module resources and updated to reflect recent changes in modeling features, user interface and application examples.

The *VPItransmissionMaker™/ VPIcomponentMaker™ User Interface Reference* provides context-sensitive help for the Photonic Design Environment. You can press `F1` in many dialogs for a description of the options available in the current context.

The *Simulation Guide* provides detailed information on the simulation process, and the *Developer Guide* explains how to extend the built-in capabilities of the Photonic Design Environment via custom macros, simulation scripts, and third-party cosimulation tools.

The *Photonic Modules Reference* (linked to each module for online help) has been extended with descriptions of the new modules and additional information on existing modules.

> **Note:** You can perform full text searches across the entire documentation library. To query the index in Adobe Reader, click the **Search** button on the toolbar, select **Edit > Search** from the menu, or press `CTRL+SHIFT+F`.

In addition to the extensive documentation provided with the product (over 5000 pages), we offer training seminars, university courses, an extensive publications list, and access to the VPIphotonics *Users Forum*.

New Features in Version 9.5

Previously, Release 9.5 brought many new features including:

- The **Resources** explorer has been significantly reworked to make library navigation more convenient and free the screen from unnecessary information.
  - Libraries have been reorganized into standard module library, application examples and the user workspace.
  - You can filter items shown in the standard module library depending on the product in use by the schematic that is currently in focus. The product set is requested on schematic creation or operations which require its expansion.
  - Toolbar icons provide a quick means to switch between groups, enable/disable filters, expand/collapse tree nodes and organize the user workspace.
  - The sorting of items in the **Resources** explorer and **Preview Pane** can be adapted in several ways.
- The PDE offers a new panel, the **Object Browser**, to help navigate more easily through deeply nested structures of your schematics. It displays the content of

currently open schematics in a hierarchical tree view and allows accessing directly the **Parameter Editor**. The information pane below the hierarchical tree provides details about the selected item.

- When dragging a new module instance to the schematic topology editor you can automatically connect it to instances that are already on the schematic by moving their ports close together.
- You can create a galaxy by selecting part of your schematic and executing the **Create Regular Galaxy** operation from the context menu.
- An empty galaxy can be created via the context menu of folders, using the **New** button menu on the **Home** ribbon, or a similar item from the **File** menu.
- When displaying a parameter value defined by a Python expression on the topology, `[Py]` is shown next to the value to emphasize that the Python interpreter is applied to this expression (and not the standard Tcl interpreter).
- Several new modules have been introduced and existing ones enhanced to support the design, analysis and optimization of multimode fiber applications. Among the new capabilities are:
  - The ***FiberMM*** module has been enhanced to simulate a variety of physical effects caused by the Kerr nonlinearity in multimode transmission fibers. The nonlinear parameters can be either specified directly or calculated automatically from the nonlinear index and transversal index profile using the mode solver in the ***SolverFiberMM*** module.
  - The new **Set Multimode Fiber Type** macro allows to set parameters of the ***SolverFiberMM*** modules to simulate typical OM1, OM2, OM3, or OM4 fibers, and of the ***SolverFiberMeasuredMM*** module to simulate typical commercially available fibers with step or graded index profiles supporting two or four modes.
  - Binary Zemax® beam files (`.zbf`) can be used to describe an optical beam and calculate the coupling coefficients between this input beam and a multimode fiber or waveguide.
  - The ***FiberMM*** module can support user-defined mode groups in addition to the already available predefined groups, which extends the range of supported fiber designs.
  - The spectral dependence of the fiber index profile in the ***SolverFiberMM*** module can be specified in more detail, which can significantly improve DGD calculations.
  - The ***AmpSysOptMM*** module as been enhanced to support the individual definition of gain and noise characteristics for different spatial modes.
- Several new modules have been introduced and existing ones enhanced to support the design, analysis and optimization of digital coherent systems applications. Among the new capabilities are:
  - The new ***Tx_PolMux_DAC*** module allows performing digital normalization, digital or analog pre-distortion, analog pulse-shaping and DAC output response emulation.
  - The new modules ***MxExtract*** and ***MxStack*** provide a convenient way for extracting or combining matrix operations that are often used for signal handling in 4D and polarization-multiplexing transmission applications.

  - Version 9.5 is compatible with version 2.0 of the DSP Library providing new data-aided frequency domain equalization, and means for off-line processing of experimental data.
  - De-skewing and digital back propagation algorithms have been added to the ***DSP_Lib*** module. The stability of the TDE-MIMO algorithm has been improved.
  - The ***ADC*** module can now estimate their effective number of bits (ENoB) in presence of distortions such as jitter and differential nonlinearities.
- The ***PhotonicsTLM*** module has been enhanced to support user-defined discretization of device sections which do not contain Bragg gratings:
  - The parameter **`NumberOfSubsections`** allows to define the maximum number of TLM subsections into which device sections are divided.
  - The parameter **`FractionalDelayFilter`** controls the use of fractional delay filters for the device sections which apply the new discretization approach.
- The ***Waveguide*** module has been extended to support the **`LogicalDelay`** switch parameter, which works in the same way as in the passive *PIC Elements* modules.
- The new ***RIN_Analyzer*** module supports the automated characterization of the relative intensity noise in lasers. It allows to measure different forms of RIN, skip transient laser dynamics, calculate the average RIN, and estimate the frequency of maximum averaged RIN.
- The new ***CompareSignals*** module compares two input signals with the specified tolerance and delivers information about their difference.
- The nonlinear transmission characteristic of the ***ComponentNL_Opt*** and ***ComponentNL_El*** modules can now be specified using Python expressions.
- The *Lab Equipment* module ***Tek_AWG_70002A*** has been extended to support not only sockets to communicate with the device, but also the VXI-11 protocol.
- A number of new features have been added to *VPIphotonicsAnalyzer* applications improving data visualization and analysis functions:
  - The support of multimode signals has been significantly enhanced. The user interface controls allow an easy selection of the required modes at different input ports. The modal content of the signal is shown in the legend of the chart.
  - The new signal analysis mode **Signal Structure** allows you to inspect the structure and properties of the optical/electrical signal data objects. This replaces the capabilities previously provided by the ***DebugInfoOutputTk*** module.
  - Signal analysis has been equipped with an embedded optical filter. It can be accessed and tuned using the controls of the **Optical Filter** tab on the **Common Settings** panel.
  - If data fitting is used in the numerical analysis, the x-axis value range can be specified for the fitting trace independently from the original data.
  - **Plot/Text** View can be switched directly from the context menu.
  - Trace legends can be hidden or made visible directly from the context menu.
  - Different annotations can be placed for different iterations, runs and sweeps.
  - Showing VPA notifications in the system tray can be switched on/off.
  - The Print command can print all charts within the VPA environment together as a single screen shot in addition to the previously available printing options.

- Information about the execution iterations of modules (`SweepInfo`) is available in all supported cosimulation interfaces.
- It is possible to use expressions specifying the bus width for wires connecting the module ports.
- Schematic parameters are available in the "init.py" initialization scripts. For function parameters, you can inspect and reset the expression type.
- The SciPy library is included on the distribution CD and installed automatically.
- Existing demonstrations have been improved and new demonstrations added.
- The documentation has been restructured to better separate task-oriented information from reference material.

Open Issues

For a list of open issues in VPItransmissionMaker / VPIcomponentMaker Version 9.7, please refer to the `OpenIssues.pdf` document included on the distribution medium (CD).

Support and Customer Feedback

We hope that you enjoy this product. We look forward to your comments and suggestions for future improvements of VPItransmissionMaker and VPIcomponentMaker.

Please contact us at support@VPIphotonics.com.