Link for ModelSim®
Release Notes

Contents

Summary by Version

This table provides quick access to what's new in each version. For clarification, see "About Release Notes" on page 1, below.

Version (Release)	New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Latest Version V2.3 (R2007b)	Yes Details	Yes Summary	Bug Reports	Printable Release Notes: PDF Current product documentation
V2.2 (R2007a)	Yes Details	No	Bug Reports	No
V2.1 (R2006b)	Yes Details	No	Bug Reports	No
V2.0 (R2006a)	Yes Details	No	Bug Reports	No
V1.4 (R14SP3)	Yes Details	No	Bug Reports	No
V1.3.1 (R14SP2)	No	No	Bug Reports	No
V1.3 (R14SP1+)	Yes Details	Yes Summary	No bug fixes	No
V1.2 (R14SP1)	Yes Details	Yes Summary	No bug fixes	No

About Release Notes

Use release notes when upgrading to a newer version to learn about new features and changes, and the potential impact on your existing files and practices. Release notes are also beneficial if you use or support multiple versions.

If you are not upgrading from the most recent previous version, review release notes for all interim versions, not just for the version you are installing. For example, when upgrading from V1.0 to V1.2, review the New Features and Changes, Version Compatibility Considerations, and Bug Reports for V1.1 and V1.2.

New Features and Changes

These include

- New functionality
- Changes to existing functionality
- Changes to system requirements (complete system requirements for the current version are at the MathWorks Web site)
- Any version compatibility considerations associated with each new feature or change

Version Compatibility Considerations

When a new feature or change introduces a reported incompatibility between versions, its description includes a **Compatibility Considerations** subsection that details the impact. For a list of all new features and changes that have reported compatibility impact, see the "Compatibility Summary for Link for ModelSim®" on page 37.

Compatibility issues that are reported after the product has been released are added to Bug Reports at the MathWorks Web site. Because bug fixes can sometimes result in incompatibilities, also review fixed bugs in Bug Reports for any compatibility impact.

Fixed Bugs and Known Problems

MathWorks Bug Reports is a user-searchable database of known problems, workarounds, and fixes. The MathWorks updates the Bug Reports database as new problems and resolutions become known, so check it as needed for the latest information.

Access Bug Reports at the MathWorks Web site using your MathWorks Account. If you are not logged in to your MathWorks Account when you link

to Bug Reports, you are prompted to log in or create an account. You then can view bug fixes and known problems for R14SP2 and more recent releases.

The Bug Reports database was introduced for R14SP2 and does not include information for prior releases. You can access a list of bug fixes made in prior versions via the links in the summary table.

Related Documentation at Web Site

Printable Release Notes (PDF). You can print release notes from the PDF version, located at the MathWorks Web site. The PDF version does not support links to other documents or to the Web site, such as to Bug Reports. Use the browser-based version of release notes for access to all information.

Product Documentation. At the MathWorks Web site, you can access complete product documentation for the current version and some previous versions, as noted in the summary table.

Version 2.3 (R2007b) Link for ModelSim®

This table summarizes what's new in Version 2.3 (R2007b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary	Bug Reports	Printable Release Notes: PDF Current product documentation

Cosimulation Support for Digital Blocks within the ADvanceMS (ADMS) Environment

Link for ModelSim version 2.3 supports cosimulation with ADMS. For details on adding libraries and starting ADMS with Link for ModelSim, see "ADMS Support" in the *Link for ModelSim User's Guide*.

Solaris 32-Bit Mode Support (On 64-Bit Platforms)

Link for ModelSim is able to run the HDL simulator side of the link in 32–bit mode and the MATLAB/Simulink side in 64–bit mode, either on a single machine or in a true cross-platform mode. See "Using the Link for ModelSim Libraries" in the *Link for ModelSim User's Guide*.

Improved Cosimulation Block GUI Eases Parameter Definition

Changes to the HDL Cosimulation block mask allow you to enter port information directly into the ports table. For more, see "Ports Pane" in the HDL Cosimulation block reference.

New Setup Script Assists in Installation Debug and Runtime Configuration

Link for ModelSim provides a guided setup script (`syschecklfm`) for configuring your simulator setup. This setup works whether your have installed Link for ModelSim and MATLAB on the same machine as ModelSim or if you have installed them on different machines. This script creates a configuration file containing the location of the appropriate Link for ModelSim MATLAB and Simulink libraries. You can then include this configuration with any other calls you make using Mentor Graphics `vsim` from ModelSim. You only need to run this script once. If you plan to use the MATLAB `vsim.m` function instead, no setup is required. See.

Note The Link for ModelSim guided setup configuration/diagnostic script works only on Unix and Linux. Windows users: please see instructions in the *Link for ModelSim User's Guide* for manually creating the configuration files.

Complete Mentor Graphics-Bundled GCC Library Compatibility

The Link for ModelSim libraries are compiled using the same compilers that MATLAB is compiled with (they vary by platform) as well as the compilers that are available with ModelSim (usually some version of gcc). This is done to ensure compatibility with other C++ libraries that may get linked into the HDL simulator, including SystemC libraries. See “Using the Link for ModelSim Libraries” in the *Link for ModelSim User’s Guide*.

Block Cosimulation Until HDL Simulator Is Ready with New Link for ModelSim MATLAB Function

The Link for ModelSim MATLAB function `pingHdlSim` blocks cosimulation by not returning until the Simulink server is loaded or until a specified timeout occurs. This function is useful if you are trying to automate a cosimulation and you need to know that the Simulink server has loaded before your script continues the simulation. See pingHdlSim in “MATLAB Functions — Alphabetical List”.

Send Tcl Commands to the HDL Simulator with New Link for ModelSim MATLAB Function

`tclHdlSim` executes a Tcl command immediately on the HDL simulator using a shared or socket connection. See tclHdlSim in “MATLAB Functions — Alphabetical List”.

Suppress Error Message with New hdldaemon Property Name/Property Value Pair

`hdldaemon` propety name/property value pair "quiet", "true" suppresses printing messages to the standard queue. Errors are still shown. See hdldaemon in “MATLAB Functions — Alphabetical List”.

Limited Support for System Verilog

You can cosimulate a model using SystemVerilog and/or SystemC with MATLAB or Simulink using Link for ModelSim. Write simple wrappers around the SystemC and make sure that the SystemVerilog cosimulation

connections are to ports or signals of data types supported by Link for ModelSim.

Link and Target Products Regrouped in New Start, Help, and Demos Category

A new product category, Links and Targets, now contains all MathWorks products that link, target, or cosimulate code.

Compatibility Considerations

This change impacts you in the following ways:

- Finding and viewing these products through the MATLAB Desktop **Start** button and in the Help browser **Contents** and **Demos** panes.
- Using the `demo` command to access the product demos.

For more about this new product category, see "Demos and Help Browser Contents Now Include New Category for Links and Targets", in the *MATLAB Release Notes*.

Improved User's Guide and Reference Documentation

The Link for ModelSim User's Guide (including the Getting Started Guide) and Reference Guide have been reorganized as a result of intense technical review and documentation usability testing, resulting in product documentation that follows a realistic workflow for EDA verification and testing.

Version 2.2 (R2007a) Link for ModelSim®

This table summarizes what's new in Version 2.2 (R2007a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	No	Bug Reports	No

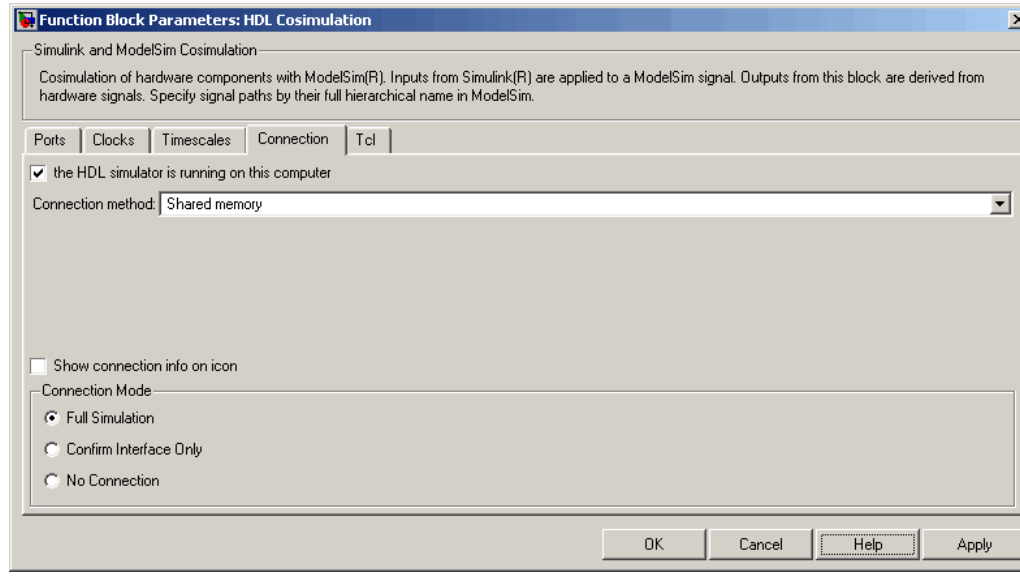
New features and changes introduced in this version are

Mixed-language (VHDL and Verilog) Cosimulation Support in Simulink Models

Link for ModelSim now supports mixed-language HDL models (models with both Verilog and VHDL components), allowing you to cosimulate VHDL and Verilog signals simultaneously.

Option to Deactivate HDL Cosimulation for Faster Simulink Model Debugging

New option panel on the Connection Pane provides the following checkboxes for bypassing the HDL simulator when running a Simulink simulation.



Select one of the following:

- **Full Simulation**: Confirm interface and run HDL simulation (default).
- **Confirm Interface Only**: Check HDL simulator for proper signal names, dimensions, and data types, but do not run HDL simulation.
- **No Connection**: Do not communicate with the HDL simulator. The HDL simulator does not need to be started.

With the 2nd and 3rd options, Link for ModelSim does not communicate with the HDL simulator during Simulink simulation.

For more about the HDL Cosimulation block, see HDL Cosimulation.

HdlServer.m for Managing Multiple HDL Simulator Connections

Link for ModelSim 2.2 contains a preliminary, beta-level scripting solution, `HdlServer.m`. As a replacement for `configuremodelsim.m` for Simulink connections, `HdlServer.m` adds capabilities for managing multiple HDL simulator connections, both remotely and locally. The script also allows for

cross-platform connections, including using MATLAB and Simulink on a Windows machine. For more information, type

```
>> help HdlServer
```

at the MATLAB prompt. If you have feedback about this beta feature, please contact The MathWorks.

Version 2.1 (R2006b) Link for ModelSim®

This table summarizes what's new in Version 2.1 (R2006b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	No	Bug Reports	No

New features and changes introduced in this version are

Support for 64-bit Linux Operating Systems Added

Link for ModelSim is now supported on 64-bit Linux operating systems.

Version 2.0 (R2006a) Link for ModelSim®

This table summarizes what's new in V2.0 (R2006a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	No	Bug Reports	No

New features and changes introduced in this version are:

Native Verilog Support

In previous releases, Link for ModelSim 2.0 interfaced to Verilog models only if VHDL wrapper code was supplied and applied using the `wrapverilog` function.

Link for ModelSim 2.0 now supports Verilog models directly, without requiring a VHDL wrapper. All Link for ModelSim MATLAB functions, and the HDL Cosimulation block, offer the same language-transparent feature set for both Verilog and VHDL models.

Note that in a mixed-language HDL model (one that contains both VHDL and Verilog components), a cosimulation block can access signals only with the language of the top-level module instance or component.

Note that the `wrapverilog` function is still supported for backward compatibility. Existing Verilog models that use `wrapverilog` will operate without disturbance.

VHDL Cosimulation Block Renamed

Cosimulation of native Verilog models is now supported. Accordingly, the VHDL Cosimulation block in the Link for ModelSim block library has been renamed HDL Cosimulation. VHDL Cosimulation blocks in existing models will continue to operate without disturbance.

Auto-Configuration of HDL Cosimulation Block Ports

The HDL Cosimulation block now lets you obtain port names and information from an HDL model under simulation in ModelSim, and automatically enter this information into the ports list of the **Ports** pane. To initiate a port information request, you use the new **Auto Fill** button in the **Ports** pane (shown in the figure in the following example).

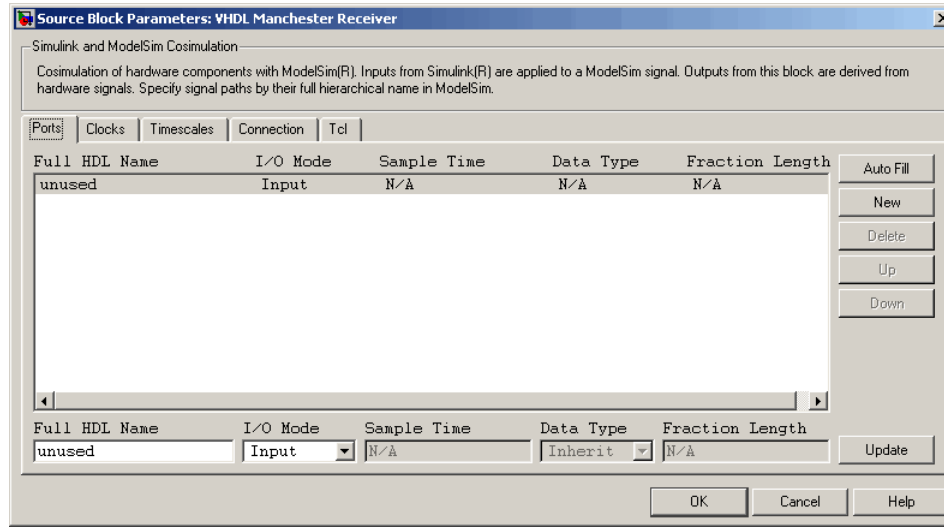
Example

To obtain port information automatically, you must first open a Simulink model containing an HDL Cosimulation block, launch ModelSim, compile and load your HDL model, and establish a connection between Simulink and ModelSim (see "Simulink and ModelSim Tutorial" if you are unfamiliar with how to set up a cosimulation.)

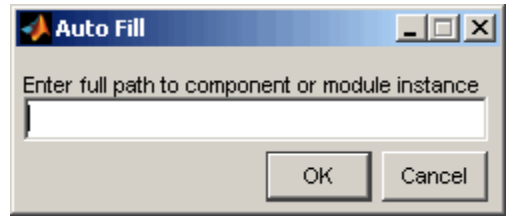
When the Simulink/ModelSim connection is established, you can initiate a ModelSim query and supply a path to a component or module in an HDL model under simulation in ModelSim. Usually, some modification of the port information is required after the query completes.

The required steps are outlined in the example below. The example is based on a modified copy of the Manchester Receiver model (see "Linking Simulink to ModelSim"), in which all signals were initially deleted from the **Ports** and **Clocks** panes.

1. Open the block parameters dialog for the HDL Cosimulation block. Click the **Ports** tab. The **Ports** pane opens.

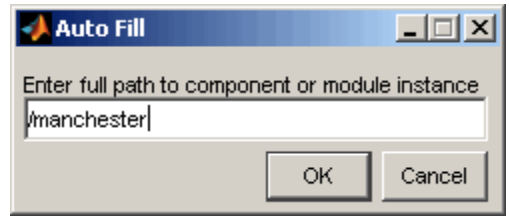


2 Click the **Auto Fill** button. The **Auto Fill** dialog opens.



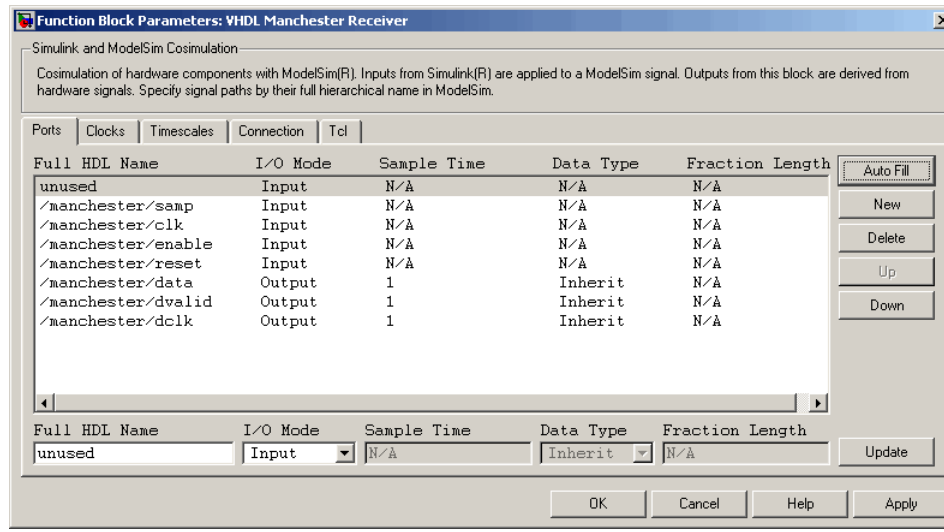
This modal dialog requests a path to a component or module in your HDL model. Enter an explicit HDL path into the edit field.

3 In this example, we will obtain port data for a VHDL component called `manchester`. The HDL path is specified as `/manchester`.



4 Click **OK**. The dialog is dismissed and the query is transmitted.

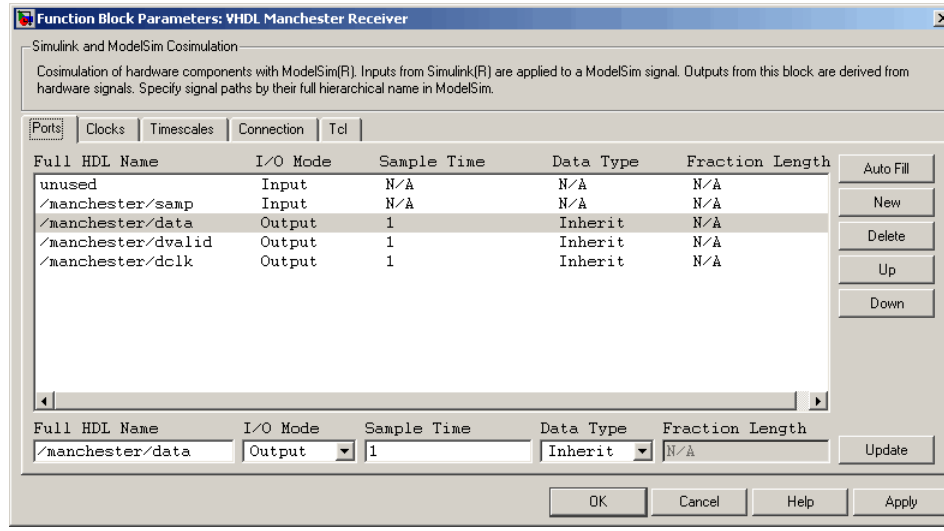
5 Port data is returned and entered into the **Ports** pane automatically, as shown in the figure below.

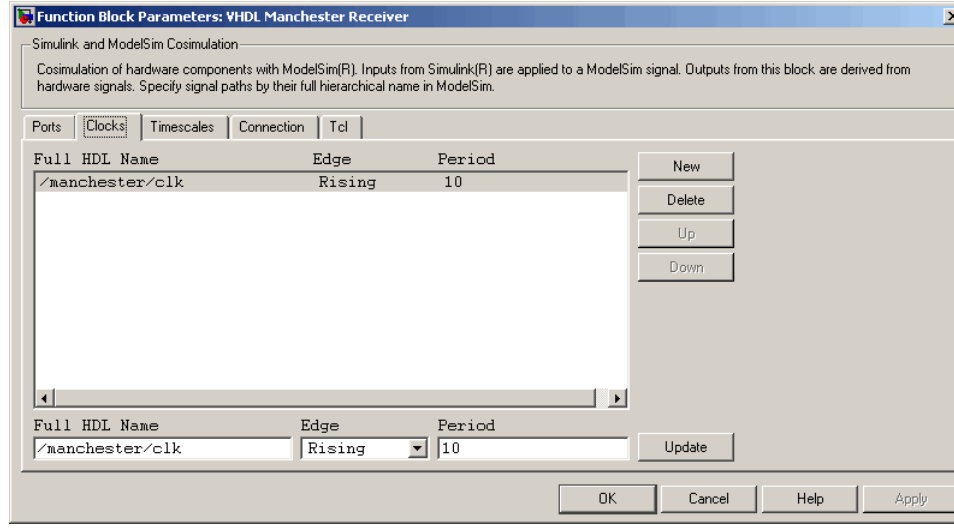


6 Click **Apply** to commit the port additions.

7 Observe that **Auto Fill** has returned information about *all* inputs and outputs for the targeted component. In many cases, this will include signals that function in ModelSim but cannot be connected in the Simulink model. You should delete any such entries from the list in the **Ports** pane.

The figure above shows that the query entered clock, clock enable, and reset ports (labeled `clk`, `enable`, and `reset` respectively) into the ports list. In this example, the `clk` signal should be entered in the **Clocks** pane, and the `enable` and `reset` signals should be deleted from the **Ports** pane, as shown in the figures below.





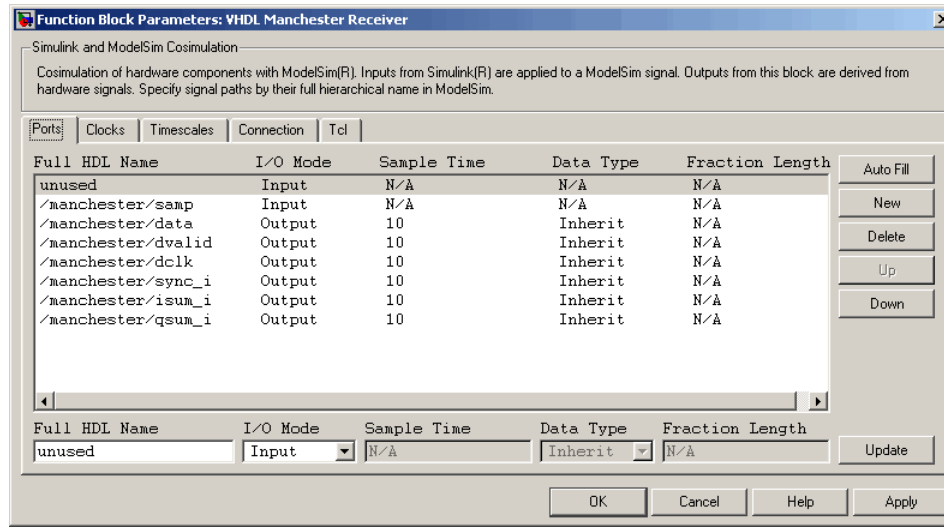
8 **Auto Fill** returns default values for output ports:

- **Sample time**: `1`
- **Data type**: `Inherit`
- **Fraction length**: `N/A`

You may need to change these values as required by your model. In this example, the **Sample time** should be set to 10 for all outputs.

9 Note that **Auto Fill** does not return information for internal signals. If your Simulink model needs to access such signals, you must enter them into the **Ports** pane manually. For example, in the case of the Manchester Receiver model, you would need to add output port entries for `/manchester/sync_i`, `/manchester/isum_i`, and `/manchester/qsum_i`, as shown below.

10 Before closing the HDL Cosimulation block parameters dialog, click **Apply** to commit any edits you have made.



HDL Cosimulation Block Supports Frame-Based Processing

The HDL Cosimulation block now supports processing of single-channel frame-based signals. Frame-based processing can improve the computational time of your Simulink models because multiple samples can be processed at once. Use of frame-based signals also lets you simulate the behavior of frame-based systems more accurately.

For a detailed description and an example of the use of frame-based processing in HDL Cosimulation, see “Using Frame-Based Processing” in the Link for ModelSim documentation.

Frame-based processing requires the Signal Processing Blockset. See “Frame-Based Signals” in the Signal Processing Blockset documentation for detailed information about frame-based processing.

inf Stop Time Support

Link for ModelSim now supports the specification of the Simulink **Stop time** parameter as `inf` . In previous releases, the total simulation time was passed to ModelSim as a 32–bit integer number of ticks. Use of `inf` stop times eliminates overflow problems for very lengthy simulations, and other problems caused by this constraint.

hdldaemon Server Passes TCL Commands to ModelSim

The `hdldaemon` server now supports transmission of TCL commands from MATLAB to ModelSim. After establishing a MATLAB to ModelSim connection through the server, you can pass a TCL command to ModelSim by including the command string in a property-value pair of the form

```
('tclcmd','command')
```

where `'command'` is a valid TCL command string.

For example, the following will display the string `This is a test` at the ModelSim prompt:

```
hldaemon('tclcmd','echo "This is a test"')
```

Note The TCL command string you specify cannot include commands that load a ModelSim project or modify simulator state. For example, they cannot include commands such as `start`, `stop`, or `restart`.

Version 1.4 (R14SP3) Link for ModelSim®

This table summarizes what's new in V1.4 (R14SP3):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	No	Bug Reports	No

New features and changes introduced in this version are:

MATLAB Component Functions

MATLAB component functions let you simulate the behavior of VHDL entities in the MATLAB environment. A MATLAB component typically provides some functionality (such as a filter) that is not yet implemented in the VHDL code.

To use a MATLAB component function, you define a stub entity (providing port definitions only) in the VHDL model. The stub entity passes its input signals to the MATLAB component function. The MATLAB component processes this data and returns the results to the outputs of the stub entity.

The programming, interfacing, and scheduling conventions for MATLAB component functions are almost identical to those for MATLAB test bench functions. The input/output arguments for a MATLAB component function are the reverse of the port arguments for a MATLAB test bench function. That is, the MATLAB component function returns signal data to the outputs of the associated VHDL entity, and receives data from the inputs of the associated VHDL entity.

The `matlabcp` function is provided in support of MATLAB component functions. `matlabcp` starts the ModelSim client component of Link for ModelSim, associates a specified instance of a VHDL entity created in ModelSim with a MATLAB function, and creates a process that schedules invocations of the specified MATLAB function.

See the Coding a MATLAB Component Function section of the Link for ModelSim documentation for information on programming conventions and a simple example function.

Version 1.3.1 (R14SP2) Link for ModelSim®

This table summarizes what's new in V1.3.1 (R14SP2):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
No	No	Bug Reports	No

Version 1.3 (R14SP1+) Link for ModelSim®

This table summarizes what's new in V1.3 (R14SP1+):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	No bug fixes	No

New features and changes introduced in this version are:

User-Defined Simulink and ModelSim Timing Relationship for Cosimulation

Overview

Link for ModelSim 1.3 lets you define the timing relationship between Simulink and ModelSim during cosimulation. Using the new **Timescales** pane of the VHDL Cosimulation block, you can now overcome problems caused by differences in the representation of simulation time between ModelSim and Simulink.

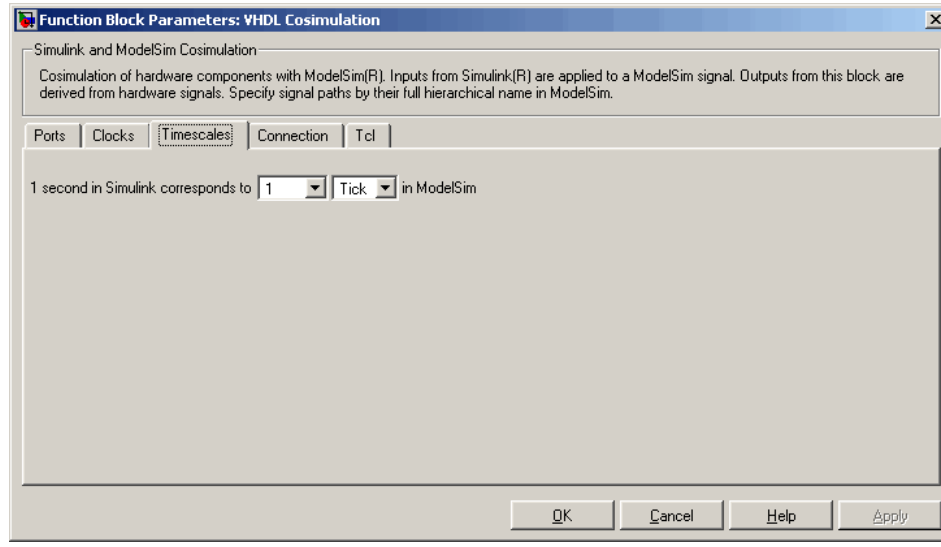
In ModelSim, the unit of simulation time is referred to as a *tick*. The duration of a tick is defined by the ModelSim *resolution limit*. The default resolution limit is 1 ns. In Simulink, simulation time is represented as a double-precision value scaled to seconds. This representation accommodates continuous models and discrete controllers.

In previous releases, the VHDL Cosimulation block supported only a fixed correspondence between simulation time in Simulink and ModelSim. In the older timing mode, one time step in Simulink corresponded to one tick in ModelSim. For example, if the total simulation time in Simulink were specified as 100 time steps, then the ModelSim VHDL simulation would run for exactly 100 ticks (i.e., `100 ns` at the default resolution limit).

New Timing Modes

Link for ModelSim 1.3 continues to support the older timing model as a default. However, the new **Timescales** pane of the VHDL Cosimulation block lets you specify the relationship between timestep sizes in a Simulink/ModelSim cosimulation with much more control and flexibility.

The figure below shows the default settings of the **Timescales** pane.



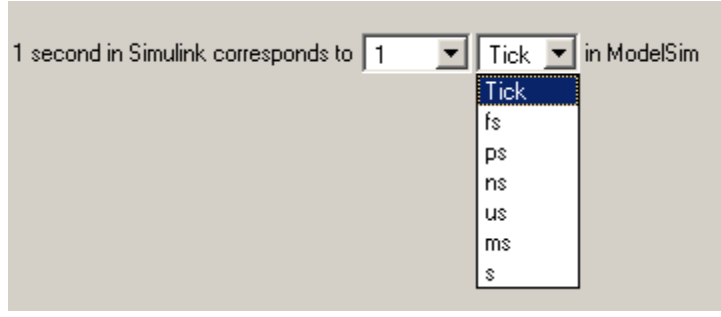
The **Timescales** pane specifies a correspondence between one second of Simulink time and some quantity of ModelSim time. This quantity of ModelSim time can be expressed in one of the following ways:

- In *relative* terms (i.e., as some number of ModelSim ticks). In this case, the cosimulation is said to operate in *relative timing mode*. In relative timing mode, *one second* in Simulink corresponds to *N ticks* in ModelSim, where `N` is a scale factor.

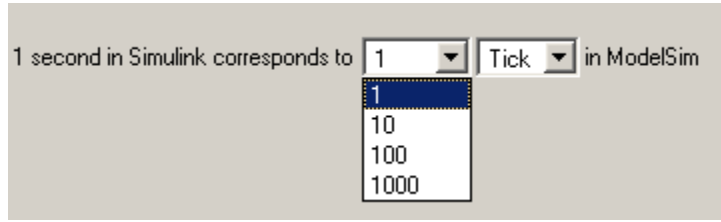
  Relative timing mode is the default.

- In *absolute* units (such as milliseconds or nanoseconds). In this case, the cosimulation is said to operate in *absolute timing mode*. In absolute timing mode, *one second* in Simulink corresponds to *(N * Tu) seconds* in ModelSim, where *Tu* is an absolute time unit (e.g., ms, ns, etc.) and *N* is a scale factor.

The **Timescales** pane contains two lists that let you select the timing mode or time unit and the scale factor. The list on the right specifies the timing mode or the time unit (see the figure below). To choose relative mode, select `Tick`. To choose absolute mode, select one of the available time units (`fs`, `ps`, `ns`, `us`, `ms`, or `s`).

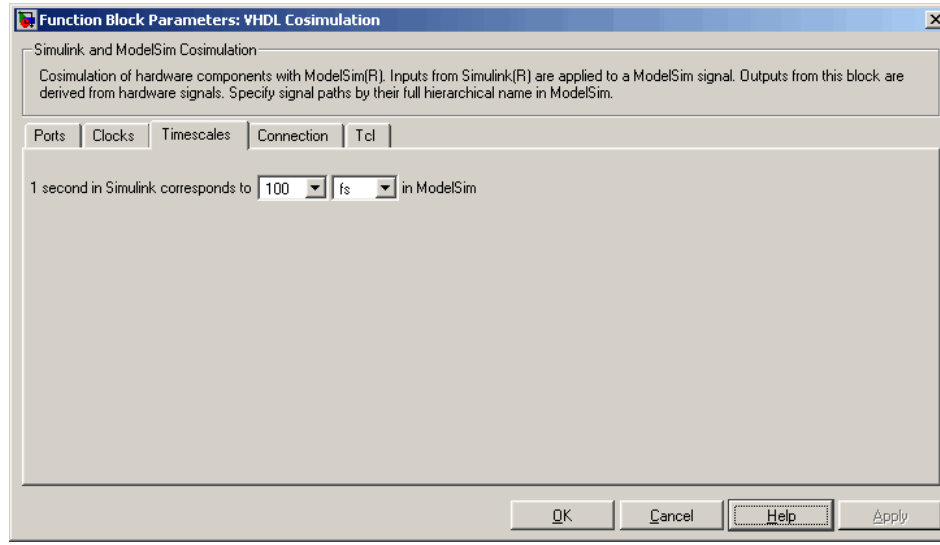


The list on the left specifies the scale factor applied to the time unit (see the figure below).



The default **Timescales** settings (see above) specify relative mode with a scale factor of 1. This default Simulink / ModelSim timing relationship is the same as the relationship defined in previous releases. The default ensures backward compatibility for existing models.

In the figure below, the **Timescales** parameters are configured for absolute mode. An absolute time unit (`fs`) and a scale factor of 100 are selected. During cosimulation, one second in Simulink corresponds to `10 fs` in ModelSim.



Representation of Simulation Time in the Link for ModelSim documentation gives a detailed description of the **Timescales** pane and the supported timing modes, with cosimulation examples.

ModelSim 6.0 Supported

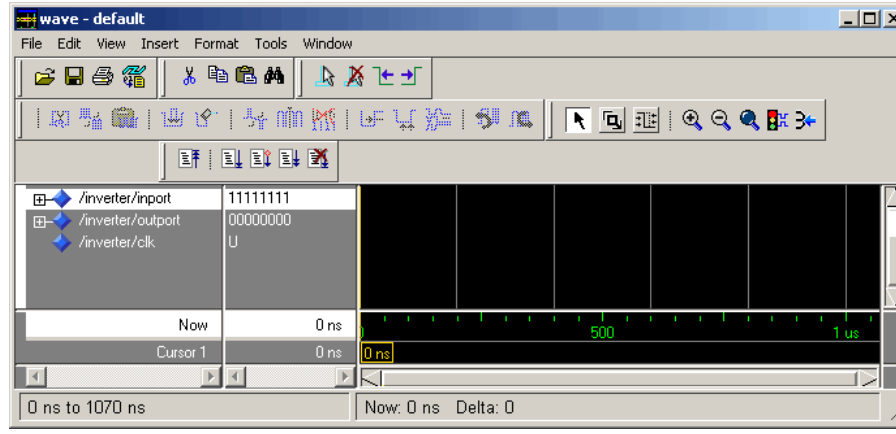
Link for ModelSim now supports ModelSim Version 6.0.

Smart Copy of Signal Names from ModelSim Wave Window

You can now copy HDL signal names (including the full HDL signal path) from the ModelSim **wave** window and paste them directly into the **Full HDL Name** field of the **Ports** or **Clocks** pane of the VHDL Cosimulation block. This convenience can save you time and errors when cosimulating an HDL design that includes long or complex signal pathnames.

To copy and paste a signal name:

1. Activate ModelSim. Select the desired signal from the signal list in the ModelSim **wave** window. In the figure below, the signal `/inverter/inport` is selected.



2. Right-click on the selected signal. Then select **Copy** from the context menu.

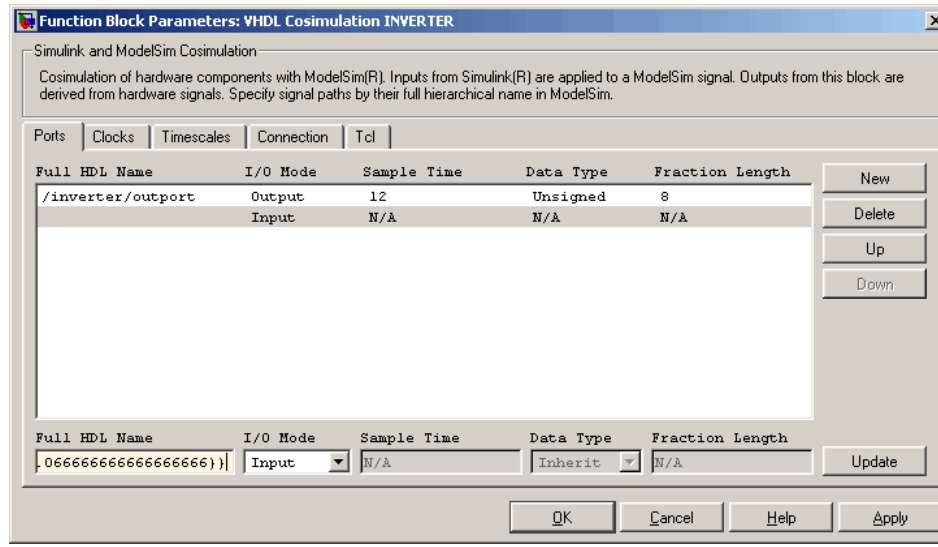
3. Activate Simulink. Then open the block parameters dialog for the desired VHDL Cosimulation block in your model.

4. Activate the appropriate (**Ports** or **Clocks**) pane of the VHDL Cosimulation block.

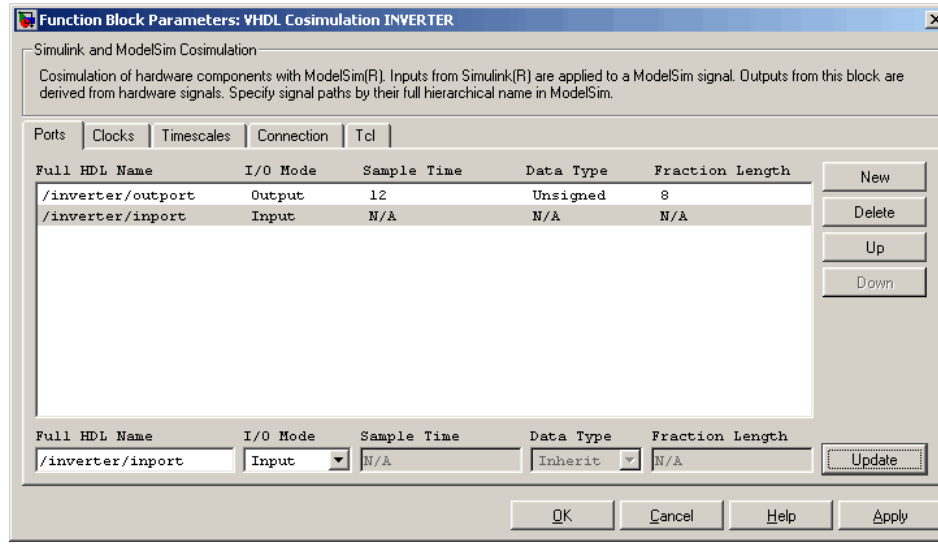
5 Select the desired signal entry from the signal list, or click the **New** button to create a new entry.

6 Select the **Full HDL Name** field.

7 Right-click and select **Paste** from the context menu to paste the signal name into the **Full HDL Name** field. At this point, the signal name is in a special clipboard format (shown below).



8 Click **Update**. Link for ModelSim translates the signal name into its final format (in this example, `/inverter/inport`) and updates the signal list.



9 If required, configure other parameters of the signal.

10 Click **Apply** when you have finished entering signal data.

VHDL Cosimulation Block No Longer Supports Use of -1 as a Block Output Port Sample Time or Clock Period

The VHDL Cosimulation block no longer supports use of -1 block output port sample time or clock period.

In previous releases, you could assign the default value -1 as

- The sample time for VHDL Cosimulation block output ports
- The clock period for time for VHDL Cosimulation block clocks

When this default was assigned, Simulink set the port sample time or the clock period equal to the fastest sample time used in the block.

Compatibility Considerations

You must explicitly specify sample times for all VHDL Cosimulation block output ports and clock periods, or accept default values. Default values are

- `1` for output port sample times
- `2` for clock periods

VHDL Cosimulation blocks in existing models should be modified to specify explicit output port sample times and clock periods. Use of the value `-1` will cause an error at simulation time.

VHDL Source and Sink Blocks Removed

The VHDL Source and VHDL Sink blocks have been removed from the Link for ModelSim block library. These blocks were simply VHDL Cosimulation blocks that were preconfigured with only output ports (Source block) or only input ports (Sink block).

Compatibility Considerations

Existing models that use VHDL Source and VHDL Sink blocks will continue to operate correctly, using the Simulink block forwarding mechanism. However, we recommend that you change existing models to use VHDL Cosimulation blocks rather than VHDL Source and VHDL Sink blocks.

setupmodelsim Command Renamed to configuremodelsim

The `setupmodelsim` command has been renamed to `configuremodelsim`. The two commands are functionally identical.

Compatibility Considerations

For backward compatibility, the `setupmodelsim` command continues to work in this release. However, we recommend that you replace `setupmodelsim` in your scripts, using `configuremodelsim` instead.

Version 1.2 (R14SP1) Link for ModelSim®

This table summarizes what's new in V1.2 (R14SP1):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary	No bug fixes	No

New features and changes introduced in this version are:

- "VHDL Cosimulation Block Enhancements" on page 30
- "Support for MATLAB/ModelSim Sessions Between Platforms of Differing Byte Ordering" on page 35

VHDL Cosimulation Block Enhancements

We have made major enhancements and revisions to the functionality and the appearance of the VHDL Cosimulation block. This release note summarizes these changes.

Per-Port Sample Time Specification for Outputs Supported

You can now specify an independent sample time for each output port on a VHDL Cosimulation block. Using the **Ports** pane of the VHDL Cosimulation block parameters dialog (see "Ports Pane" on page 31) you can specify an explicit sample time, or specify a default (`-1`). In the default case, Simulink sets the sample time to the fastest sample time used in the block.

Per-Port Data Type Specification for Outputs Supported

You can now force fixed point data types on individual output ports of a VHDL Cosimulation block, using the **Ports** pane of the VHDL Cosimulation block parameters dialog (see "Ports Pane" on page 31). By default, Simulink determines the data type by back-propagation or by querying ModelSim.

Alternatively, you can assign an explicit data type (with optional fraction length) using the **Data Type** and **Fraction length** fields.

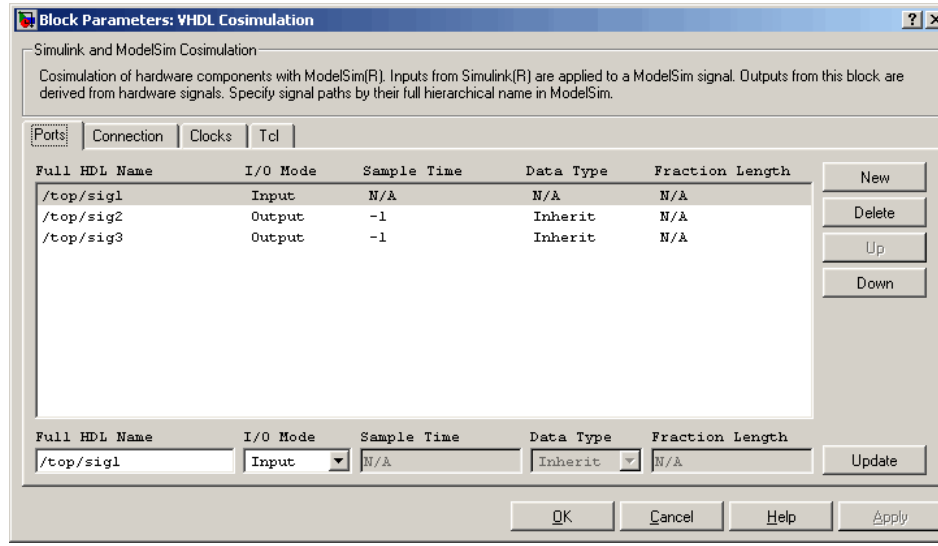
Specification of Independent Clock Sample Times Supported

Using the **Clocks** pane of the VHDL Cosimulation block parameters dialog (see "Clocks Pane" on page 33). you can now specify period of each clock in the model explicitly, or specify `-1` to use a default value supplied by Simulink. In the default case, Simulink sets the clock period to the fastest sample time used in the block.

Improved and Revised VHDL Cosimulation Block Parameters Dialog Box

The sections below illustrate and summarize the improvements that have been made to the VHDL Cosimulation block GUI.

Ports Pane. The figure below shows the revised layout of the **Ports** pane of the VHDL Cosimulation Block Parameters dialog box.



The **Ports** pane now displays a scrolling list of VHDL signals corresponding to ports on the VHDL Cosimulation block. The buttons to the right of the list

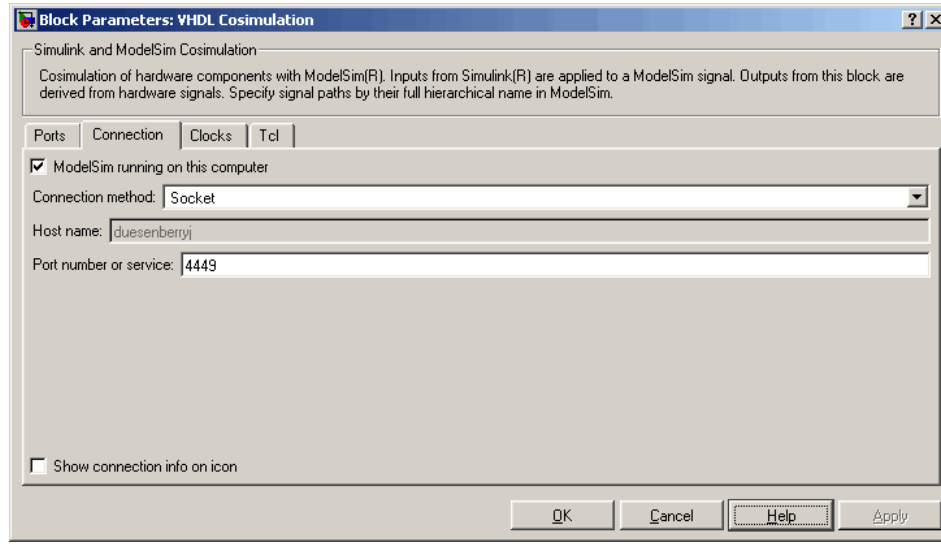
let you add, delete, or reposition signals in the list. To set the properties of a signal, select the desired signal from the list and enter values into the property fields below the list.

The **Ports** pane supports the following properties and capabilities:

- The **Full HDL Name** field lets you enter the VHDL pathname for a signal anywhere in the hierarchy of the VHDL model.
- The **I/O Mode** menu lets you select whether a signal is associated with an input or output port.
- The **Sample Time** field lets you specify a sample time, per port, for outputs.
- The **Data Type** and **Fraction length** fields let you specify a fixed point data type for individual output ports of a VHDL Cosimulation block.

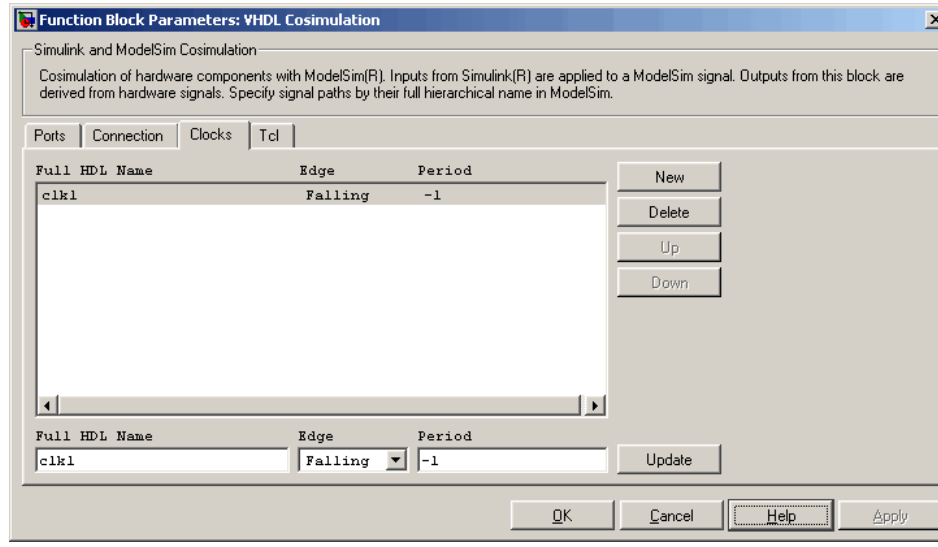
Connection Pane. The figure below shows the default layout of the **Connection** pane (formerly labelled as the **Comm** pane) of the VHDL Cosimulation Block Parameters dialog box.

By default, the block is configured for shared memory communication. If you select TCP/IP socket mode communication, the pane displays additional properties, as shown in the figure below.



When the new **Show connection info on icon** option is selected, information about the selected communication method and (if applicable) communication options is displayed on the VHDL Cosimulation block icon in the Simulink model.

Clocks Pane. The figure below shows the default layout of the **Clocks** pane of the VHDL Cosimulation Block Parameters dialog box.

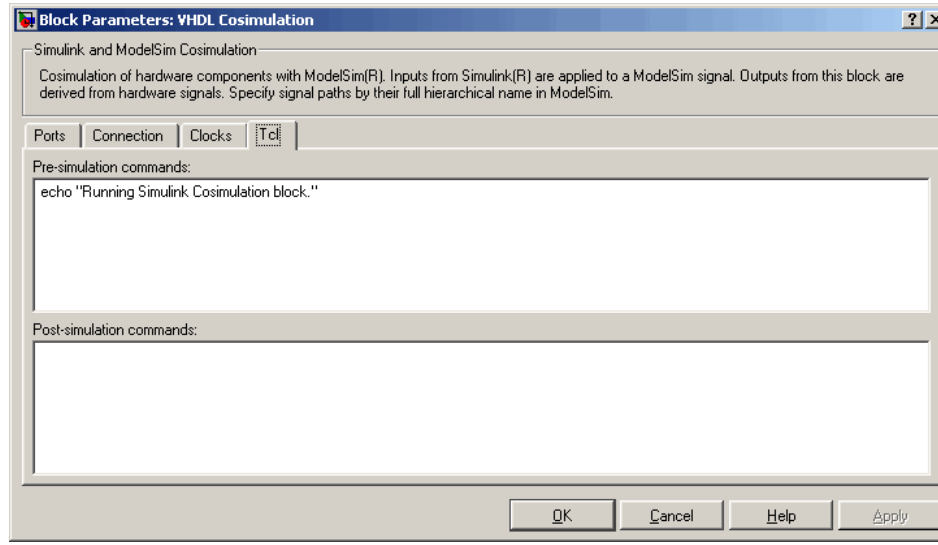


The **Clocks** pane now displays a scrolling list of VHDL clock signals The buttons to the right of the list let you add, delete, or reposition clock signals in the list. To set the properties of a clock signal, select the desired signal from the list and enter values into the property fields below the list.

The **Clocks** pane supports the following properties and capabilities:

- The **Full HDL Name** field lets you enter the VHDL pathname for a clock signal.
- The **Edge** menu lets you specify either a rising-edge clock or a falling-edge clock.
- The **Period** field lets you specify the clock period explicitly, or specify `-1` to use a default value supplied by Simulink.

Tcl Pane. The figure below shows the revised layout of the **Tcl** pane of the VHDL Cosimulation Block Parameters dialog box.



You can now specify Tcl commands in the text boxes in one line per command format, or enter multiple commands per line by appending each command with a semicolon (;), the standard Tcl concatenation operator.

Support for MATLAB/ModelSim Sessions Between Platforms of Differing Byte Ordering

You can now run MATLAB/ModelSim sessions in TCP/IP socket mode between platforms having different byte ordering.

Compatibility Considerations

In previous releases, Link for ModelSim required that when MATLAB/ModelSim sessions were run in TCP/IP socket mode, all connected systems must support the same byte ordering (e.g., little-endian or big-endian). This restriction has been removed.

The following table illustrates the currently supported MATLAB / ModelSim connections.

MATLAB / ModelSim Platforms	PC	Linux	Solaris
PC	Yes	Yes	Yes (*new in Link for ModelSim v. 1.2*)
Linux		Yes	Yes (*new in Link for ModelSim v. 1.2*)
Solaris			Yes

Compatibility Summary for Link for ModelSim®

This table summarizes new features and changes that might cause incompatibilities when you upgrade from an earlier version, or when you use files on multiple versions. Details are provided in the description of the new feature or change.

Version (Release)	New Features and Changes with Version Compatibility Impact
Latest Version V2.3 (R2007b)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "Link and Target Products Regrouped in New Start, Help, and Demos Category" on page 7
V2.2 (R2007a)	None
V2.1 (R2006b)	None
V2.0 (R2006a)	None
V1.4 (R14SP3)	None
V1.3.1 (R14SP2)	None

Version (Release)	New Features and Changes with Version Compatibility Impact
V1.3 (R14SP1+)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "VHDL Cosimulation Block No Longer Supports Use of -1 as a Block Output Port Sample Time or Clock Period" on page 28 • "VHDL Source and Sink Blocks Removed" on page 29 • "setupmodelsim Command Renamed to configuremodelsim" on page 29
V1.2 (R14SP1)	See the **Compatibility Considerations** subheading for this new feature: • "Support for MATLAB/ModelSim Sessions Between Platforms of Differing Byte Ordering" on page 35