Link for ModelSim® Release Notes

Note No significant new features have been introduced for Version 1.3.1 of the Link for ModelSim. With the exception of bug fixes, the product is essentially unchanged from Version 1.3. The following Release Notes discuss bug fixes and important issues related to the Link for ModelSim 1.3.1.

The Link for ModelSim Release Notes also provide information about the earlier versions of the product, in case you are upgrading from an earlier version:

Contents

4 Link for ModelSim 1.1.1 Release Notes

5 Link for ModelSim 1.1 Release Notes

1

Link for ModelSim 1.3.1 Release Notes

Major Bug Fixes

The Link for ModelSim 1.3.1 includes important bug fixes made since Version 1.3. You can see a list of major bug fixes in Version 1.3.1 on the MathWorks Web site. If you are viewing these release notes in PDF form on the MathWorks Web site, click the words “bug fixes” in the sentence above to see the notes about major fixes. If you are upgrading from a version earlier than Version 1.3.1, you should also see Version 1.3 “Major Bug Fixes” on page 2-9.

Known Software and Documentation Problems

The MathWorks Web site includes a list of known software and documentation problems in Version 1.3.1. If you are viewing these release notes in PDF form on the MathWorks Web site, click the word “problems” in the sentence above to see the notes about known problems.

2

Link for ModelSim 1.3 Release Notes

New Features

This section describes new features introduced in Link for ModelSim 1.3.

User-Defined Simulink and ModelSim Timing Relationship for Cosimulation

Overview

The 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.

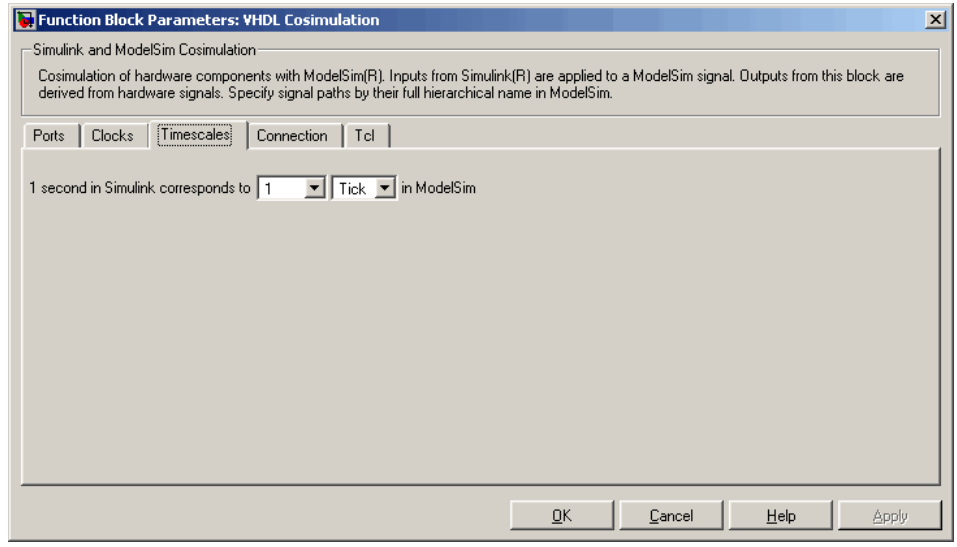
In ModelSim, the unit of simulation time is referred to as a *tick*. The duration of a tick is defined by the ModelSim *resolution limit*. The default resolution limit is 1 ns. In Simulink, simulation time is represented as a double-precision value scaled to seconds. This representation accommodates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

The 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.



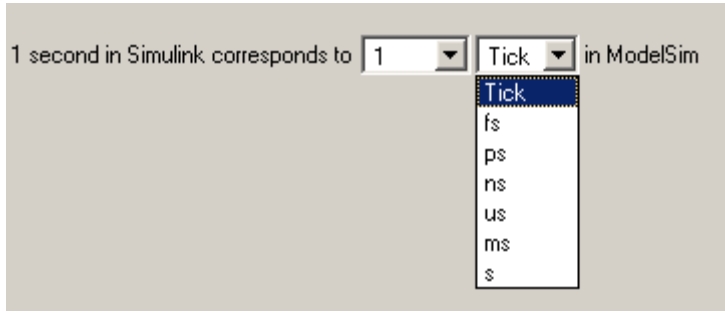
The **Timescales** pane specifies a correspondence between one second of Simulink time and some quantity of ModelSim time. This quantity of ModelSim time can be expressed in one of the following ways:

- 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.

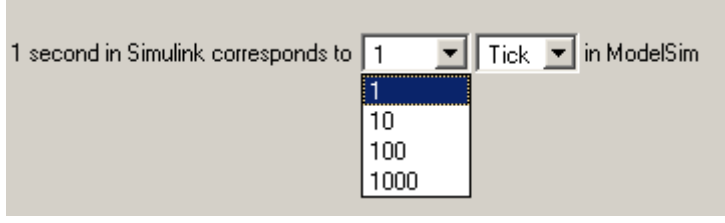
  Relative timing mode is the default.

- 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`).

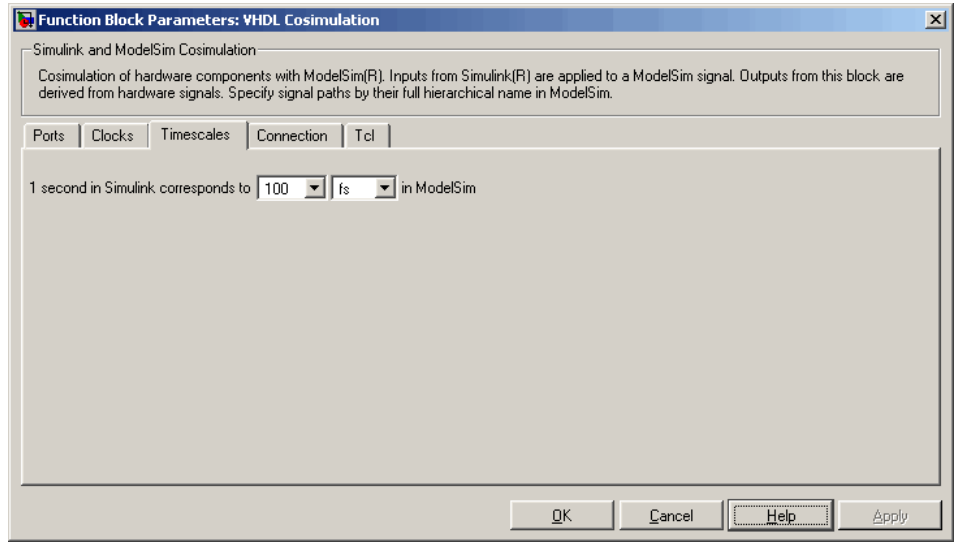


The list on the left specifies the scale factor applied to the time unit (see the figure below).



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

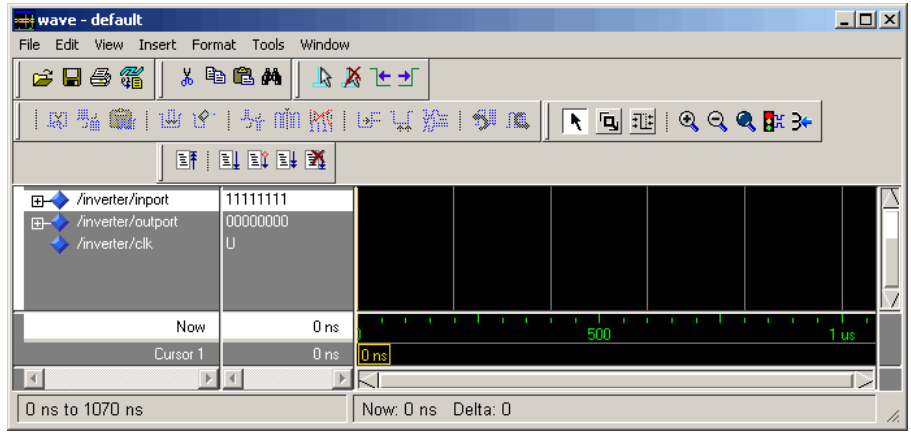
The Link for ModelSim now supports ModelSim Version 6.0.

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 Path** 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.

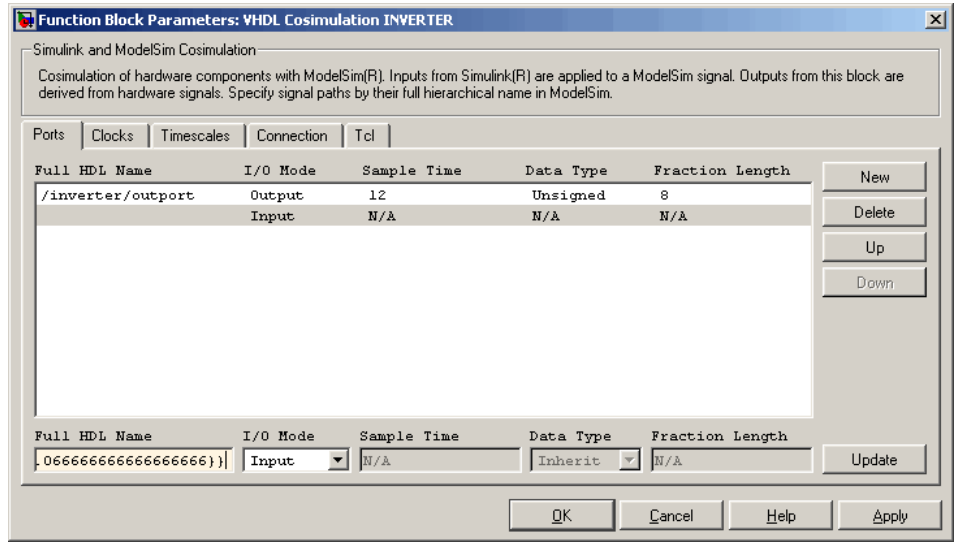
3 Activate Simulink. Then open the block parameters dialog for the desired VHDL Cosimulation block in your model.

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

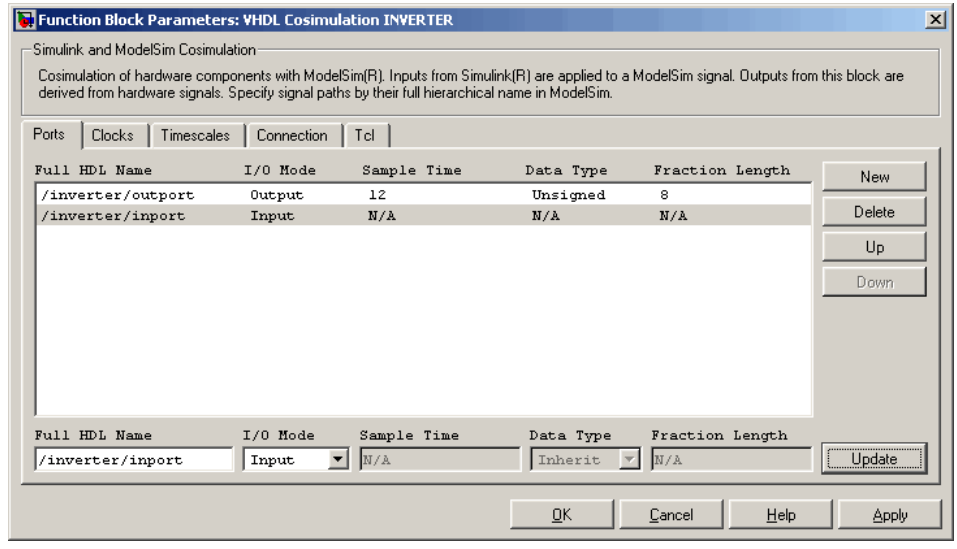
5 Select the desired signal entry from the signal list, or click the **New** button to create a new entry.

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

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



8 Click **Update**. The 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.

Major Bug Fixes

The Link for ModelSim 1.3 includes several bug fixes made since Version 1.2. This section includes a link to a description of major bug fixes in Version 1.3.

If you are viewing these Release Notes in PDF form, please refer to the HTML form of the Release Notes, using either the Help browser or the MathWorks Web site and use the link provided.

Upgrading from a Previous Release

VHDL Cosimulation Output Port Sample Times and Clock Periods Must Be Specified Explicitly

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.

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

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.

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.

3

Link for ModelSim 1.2 Release Notes

New Features

This section describes new features introduced in Link for ModelSim 1.2.

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. For a full description of the VHDL Cosimulation block, see the VHDL Cosimulation block reference in the online Link for ModelSim documentation.

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 3-3) 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 3-3). 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.

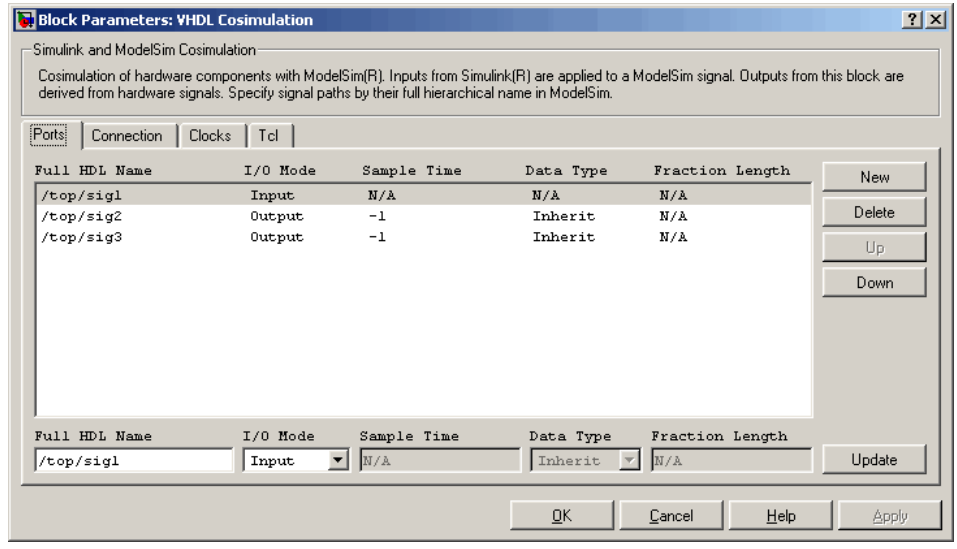
Specification of Independent Clock Sample Times Supported

Using the **Clocks** pane of the VHDL Cosimulation block parameters dialog (see “Clocks Pane” on page 3-5). 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.



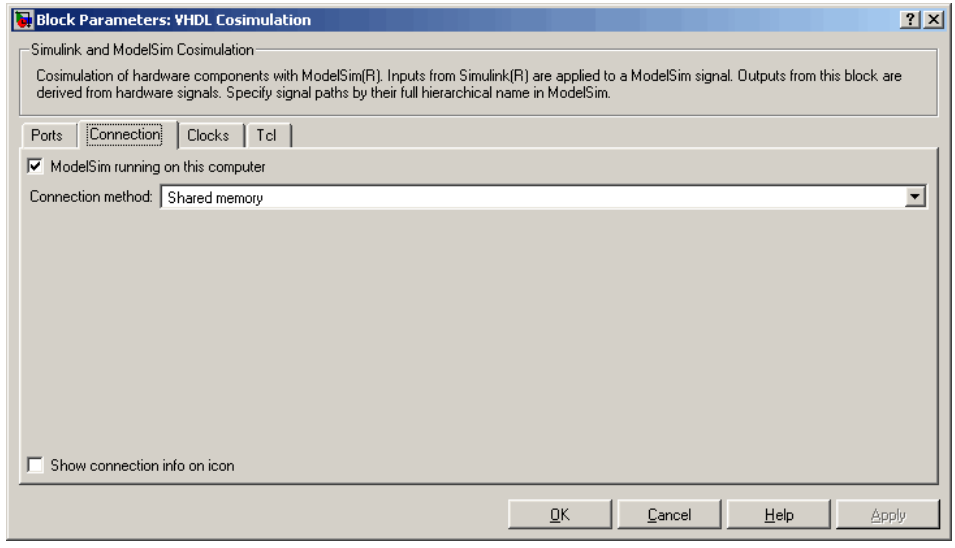
The **Ports** pane now displays a scrolling list of VHDL signals corresponding to ports on the VHDL Cosimulation block. The buttons to the right of the list let you add, delete, or reposition signals in the list. To set the properties of a signal, select the desired signal from the list and enter values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

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

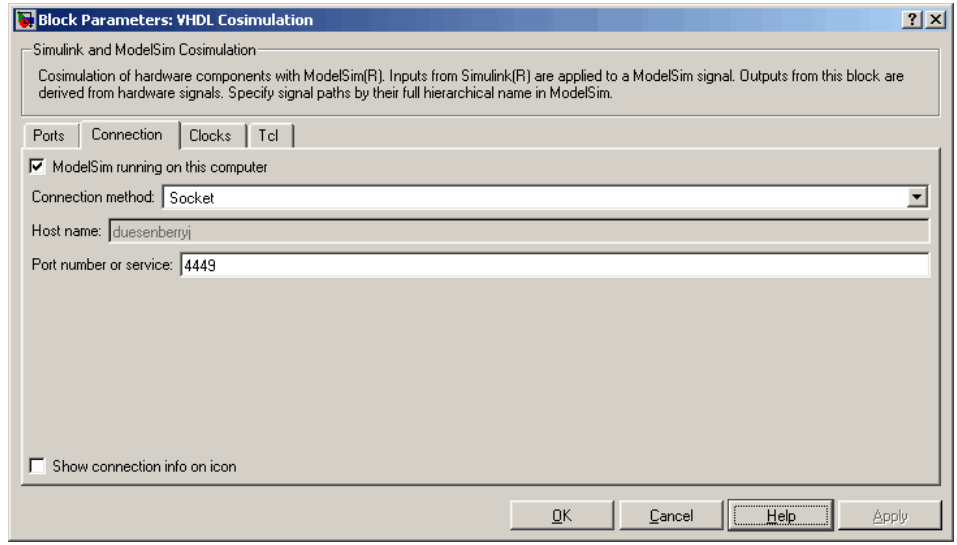
- The **Full HDL Name** field lets you enter the VHDL pathname for a signal anywhere in the hierarchy of the VHDL model.
- The **I/O Mode** menu lets you select whether a signal is associated with an input or output port.
- The **Sample Time** field lets you specify a sample time, per port, for outputs.

- 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.

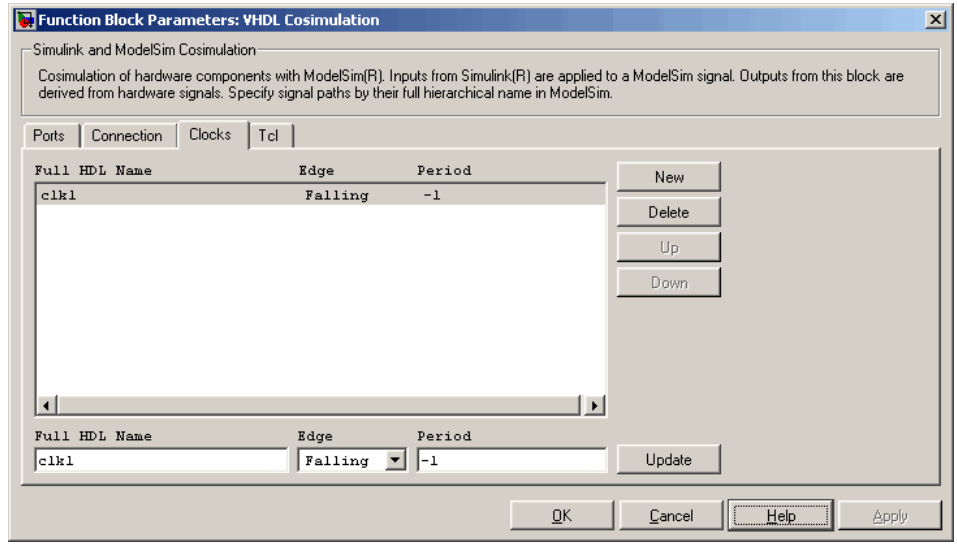


By default, as shown above, the block is configured for shared memory communication. If you select TCP/IP socket mode communication, the pane displays additional properties, as shown in the figure below.



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.

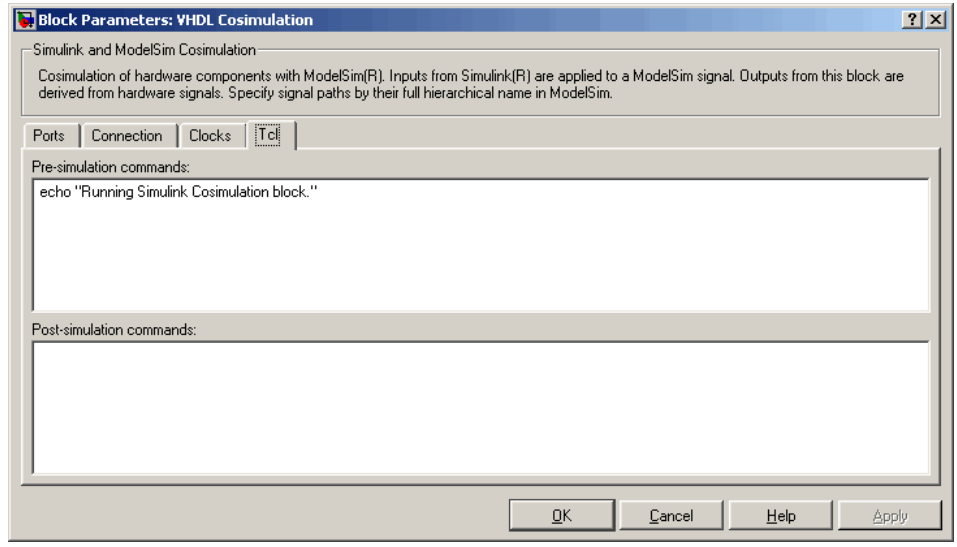


The **Clocks** pane now displays a scrolling list of VHDL clock signals The buttons to the right of the list let you add, delete, or reposition clock signals in the list. To set the properties of a clock signal, select the desired signal from the list and enter values into the property fields below the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

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

Major Bug Fixes

The Link for ModelSim 1.3 includes several bug fixes made since Version 1.1. This section includes a link to a description of major bug fixes in Version 1.2.

If you are viewing these Release Notes in PDF form, please refer to the HTML form of the Release Notes, using either the Help browser or the MathWorks Web site and use the link provided

4

Link for ModelSim 1.1.1 Release Notes

Upgrading from an Earlier Release

This section describes upgrade issues involved in moving to Link for ModelSim 1.1.1 from Version1.1 and earlier.

Initialization Required for Test Bench Function Output Parameters

This release note applies to users who have written, or plan to write, MATLAB test bench functions for use with Link for ModelSim.

Under MATLAB 7.0 (Release 14), code validation performed on callback functions is more stringent than in prior releases. Test bench functions must now initialize their output parameters (`iport` and `tnext`), even if these parameters are not actually referenced otherwise in the code. If the output parameters are not initialized correctly, warnings will occur when the test bench function executes.

This requirement can be met by initially setting the output parameters to empty values, as in the following code excerpt:

```
function [iport, tnext] = MyFunctionName(oport, tnow, portinfo)

tnext = [];
iport = struct();
...
```

It is recommended practice to locate this code at the beginning of your function.

If you created test bench functions for an earlier release of MATLAB, you should check and (if necessary) modify your code to make sure that it initializes output parameters as described above.

For further information about test bench functions and their parameters, see " Setting up Expected Parameters" in the " Coding a MATLAB Test Bench Function" section of the Link for ModelSim online documentation.

5

Link for ModelSim 1.1 Release Notes

New Features

Linux Platform Supported

Link for ModelSim 1.1 introduces support for the Linux platform.

Major Bug Fixes

The Link for ModelSim 1.1 includes bug fixes made since Version 1.0. This section describes major bug fixes.

Most Demos Cannot Be Run by Multiple Users on Solaris

Most of the product demos create their VHDL work directory in the standard MATLAB `tempdir` (typically `/tmp`). This meant that only one user at a time could run these demos on the same computer, and no other user could run the demos on that system until the previous user cleaned up the files in the temporary directory. The following demos exhibited this problem:

```
manchestermodel
manchestermodelcommblks
multimanchester
mixedmanchester
modsimosc
```

The `modsimrand` demo did not have this problem because you can specify a unique project directory in the **Generate VHDL project in directory** field of the demo dialog.

This is now fixed in Link for ModelSim 1.1.

Use of Shared Memory Mode on Solaris May Generate Errors or Cause Link for ModelSim to Stall

On Solaris, you could encounter problems if multiple users ran MATLAB or Simulink cosimulations in shared memory mode, either simultaneously or one after the other on the same computer. For example, if one user ran a demo using shared memory one day, and the next day another user tried to run a demo on the same computer also using shared memory, the second user could encounter difficulties. You could also encounter problems if one user tried to run two simulations of the same type using shared memory simultaneously. In either case, error messages such as the following could appear:

```
Failed to connect to server
```

```
Unable to open required FIFO
```

The group access portion of this bug is now fixed in Link for ModelSim 1.1. For information on the single-user aspect of this bug, see “Use of Shared Memory Mode on Solaris or Linux May Generate Errors or Cause Link for ModelSim to Stall” on page 5-5.

Known Software and Documentation Problems

This section describes known software and documentation problems that exist in Link for ModelSim 1.1.

Link for ModelSim Installations Reset the MATLAB Path

When you install Link for ModelSim 1.0 over an existing MATLAB installation, the installation procedure resets your existing MATLAB path, potentially removing products previously installed. This is most likely to occur to third-party product or user-defined extensions to the path.

To correct this condition, modify the MATLAB path definitions to include any paths removed as a result of the reset. You can modify the path definitions by doing one of the following:

- Click **File–>Set Path** and add missing paths.
- Open the file `MATLABROOT\toolbox\local\pathdef.m` and append missing paths.

A planned change to the installer will expand rather than reset the MATLAB path.

For more information on Link for ModelSim installation and setup, see Installation and Setup.

Use of Shared Memory Mode on Solaris or Linux May Generate Errors or Cause Link for ModelSim to Stall

On systems running Solaris or Linux, you may encounter problems if you try to run two cosimulations of the same type (MATLAB or Simulink) using shared memory mode simultaneously. Error messages such as the following may appear:

```
Failed to connect to server

Unable to open required FIFO
```

It is also possible for `hdldaemon('kill')` to stall MATLAB indefinitely.

A simple work around for this problem is to use TCP/IP socket communication instead of shared memory. If this is not an option because you need the performance gain provided by shared memory, you can work around the problem by launching each cosimulation session from a separate shell with each shell defining the environment variable `TMPDIR` to a unique path.

The shared memory mode of communication creates and uses special files in an existing writable directory defined by `TMPDIR`. The problems are caused by contention for these files. You can eliminate contention and avoid possible problems by setting `TMPDIR` to a unique path before starting each cosimulation session.

For example, to start two shared memory cosimulation sessions of the same type on the same system

1. From a given shell, set the value of `TMPDIR` to a valid writable path.
2. From that same shell, start a MATLAB or Simulink cosimulation session.
3. Start another shell and set the value of `TMPDIR` for that shell to a different path.
4. From the second shell, start another cosimulation session of the type started in 2.

In this scenario, Link for ModelSim creates the special files in distinct temporary directories, preventing contention.

The following shell script illustrates one way to automatically set `TMPDIR` to a unique path. If `TMPDIR` does not exist, the script sets it to a default. Next, the script sets `TMPDIR` to the name of a subdirectory based on the user name and process ID of the shell. Finally, the script ensures that the subdirectory actually exists.

```
if ( ! $?TMPDIR )  then
  setenv TMPDIR /var/tmp
else if ( $TMPDIR == "" ) then
  setenv TMPDIR /var/tmp
endif
```

```
setenv TMPDIR $TMPDIR/${USER}$$
mkdir -p $TMPDIR
```

Notes

- If you get the error messages noted above and you are not running a cosimulation, you can delete the special files in `TMPDIR`.
- Be aware that other programs may also use `TMPDIR` to identify directories for temporary files.

MATLAB pause Function Suspends ModelSim Callbacks

When used in a MATLAB callback function associated with a VHDL entity, the MATLAB `pause` function causes MATLAB to block and wait for you to press a key. This function also blocks callbacks from ModelSim, suspending ModelSim and MATLAB communication. To continue processing, press any key while the input focus is in MATLAB. Although no data is lost, you might not expect this behavior.

Real-Time Workshop Code Generation Not Supported

The ModelSim Cosimulation blocks do not support Real-Time Workshop® code generation. If you attempt to generate code for one of the blocks with Real-Time Workshop, you will receive an error message similar to the following:

```
Error executing build command: Error using ==> make_rtw
  Error using ==> rtwgen
  S-function block 'manchestermodelcommblks/VHDL Manchester Receiver'
parameters must be numeric arrays for Real-Time Workshop.
```