Link for ModelSim Release Notes

These Release Notes discuss the following topics as they pertain to Link for ModelSim 1.1:

Contents

1 Link for ModelSim 1.1 Release Notes

Link for ModelSim 1.1 Release Notes

New Features

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 1–7.

Known Software and Documentation Problems

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

edit Command Fails in ModelSim on Solaris or Linux when ModelSim is Invoked from MATLAB

If you invoke ModelSim on a Solaris or Linux system from MATLAB with the MATLAB `vsim` function, attempts to use the ModelSim `edit` command fail and you receive the following message:

```
# forked process couldn't set up input/output: bad file number
```

This error is caused by a ModelSim limitation on the use of the `vsim` command's `-gui` option, which is used by the MATLAB `vsim` function. The same failure results if you invoke `vsim` from a UNIX shell with the `-gui` option as follows:

```
vsim -gui
```

If you then use the `edit` command in the `vsim` console window, you get the same failure as noted above.

To work around this limitation, use the ModelSim GUI to open a VHDL file for editing.

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.

VHDL Ports of Type ARRAY(0 to 0) Not Supported

Currently, Link for ModelSim does not support VHDL ports of type `ARRAY(0 to 0)`. For example, you cannot use the following type definition:

```
TYPE fake_bool is ARRAY (0 to 0) of std_logic
```

Use of a port of this type causes ModelSim to crash, potentially leaving MATLAB and the communication link in a bad or inconsistent state.

Support for this type definition is planned for a future release of the product.

For more information on supported VHDL data types, see Specifying Port Data Types.

hdldaemon Shutdown in Shared Memory Mode on Windows Requires Exiting ModelSim Simulator

If your application uses shared memory communication on a Windows platform and you try to use `hdldaemon` to shut down the MATLAB server while a ModelSim connection exists, the shutdown fails and a message similar to the following appears:

```
??? HDLDaemon server shutdown FAILED.
Make sure all connections from ModelSim are closed.
```

Close connections from ModelSim by doing one of the following:

To...	Enter the Command...
Close the connection without stopping the ModelSim simulation	`nomatlabtb`
Stop the ModelSim simulation	`quit -sim`
Exit ModelSim entirely	`quit`

Then, repeat the call to `hdldaemon`.

For more information on shutting down the MATLAB server, see Stopping a Test Bench Session.

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 (MALTAB 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.

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 means that only one user at a time can run these demos on the same computer, and no other user can run the demos on that system until the previous user cleans up the files in the temporary directory. The following demos exhibit this problem:

```
manchestermodel
manchestermodelcommblks
multimanchester
mixedmanchester
modsimosc
```

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

matlabtb and matlabtbeval Commands Display Uninformative Error Message When MATLAB server is Not Running

If you issue the `matlabtb` or `matlabtbeval` command in ModelSim when the MATLAB server is not running, the command displays the following error message:

```
MLerr - The system cannot find the file specified.
```

To check the server's status, issue a call to the `hdldaemon` function with the `'status'` option.

```
hdldaemon('status')
```

If the server is not running, the following message appears:

```
HDLDaemon is NOT running
```

To correct the condition, start the server with a call to `hdldaemon`.

For more information on using the `matlabtb` and `matlabtbeval` commands, see Initializing the Simulator for a MATLAB Test Bench Session.

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.

VHDL Cosimulation Block Parameters Dialog Tabs May Display Incorrectly on Older Versions of Solaris

If you are running a version of Solaris prior to Version 9, the tabs of the VHDL Cosimulation block parameters dialog may not display correctly. Although you will not be able to see the tab labels, the tabs should still work correctly if you click just above them. From left-to-right, the tab labels are

- Ports
- Comm
- Clocks
- Tcl

For more information, see Configuring a VHDL Cosimulation Block or contact technical support.

Client/Server Components Do Not Detect TCP/IP Port Filtering

If your client or server environment filters a TCP/IP socket port that you specify, the Link for ModelSim interface is not able to establish a connection. The server and client display messages that suggest a connection has not been made, but the messages are not explicit and do not explain why.

When you start `hdldaemon` in MATLAB, the server displays

```
HDLDaemon server started on port port#
```

When you start the simulation in ModelSim, the simulation just does not run. If you restart it, ModelSim displays the following warning:

```
#MLWarn - MATLAB server not available (yet), The entity
'entityname' will not be active
```

If you return to MATLAB and check the server status, a message displays indicating that the server is running with no connections.

```
x=hdldaemon('status')
HDLDaemon server is running on port port# with 0 connections
```

If you experience this behavior and you suspect the TCP/IP socket port you are using is being filtered, check it by clicking through the following sequence:

Start->Settings->Network and Dialup Connections->Local Area Connection->Properties->Internet Protocol (TCP/IP)->Properties->Advanced->Options->TCP/IP filtering->Properties

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.
```

Simulink Accelerator Mode Not Supported

The ModelSim Cosimulation blocks do not support the Simulink Accelerator. When cosimulating with ModelSim, Simulink models must remain in Normal mode. Support for the Accelerator is under consideration for a future release.