

VCS MX Installation Notes

Version D-2009.12

December 2009

This document contains instructions about installing the VCS MX (VHDL Simulation and Verilog Simulation) tool.

This document contains the following sections:

- [“Media Availability and Supported Platforms”](#)
- [“Accessing Memory Beyond 2 GB With 32-Bit Tools”](#)
- [“Installing the Software”](#)
- [“Setting the Environment Variables”](#)
- [“Verifying the VCS MX Installation”](#)

The instructions in this chapter also apply to VCS MXi.

Media Availability and Supported Platforms

The VCS MX tool is available by EST or on CD. Obtain the appropriate binary executable files based on the operating system you need. [Table 1-1](#) shows the supported platforms for this release.

Table 1-1 Platforms and Keywords

Platform	Operating system	Synopsys platform keyword	Additional keywords
x86_64	Red Hat Enterprise Linux v4, 5 ¹	amd64 (64-bit mode) linux (32-bit mode) ²	amd64_32
x86_64	SUSE Enterprise Linux v9, 10 ¹	suse64 (64-bit mode) suse32 (32-bit mode) ²	suse64_32
x86_64	Solaris 10	x86sol64 (64-bit mode) x86sol32 (32-bit mode)	
IBM RS6000	AIX 5.3	rs6000 (32-bit mode) aix64 (64-bit mode)	
x86	Red Hat Enterprise Linux v4, 5 ¹	linux (32-bit mode) ²	
x86	SUSE Enterprise Linux v9, 10 ¹	suse32 (32-bit mode)	
Sun SPARC	Solaris 9, 10 ¹	sparc64 (64-bit mode) sparcOS5 (32-bit mode)	sparc64_32

1. *Binary-compatible hardware platform or operating system. Note, however, that binary compatibility is not guaranteed. See <http://www.synopsys.com/products/platforms> for the latest information.*

2. *Binary-compatible hardware platform or operating system. Note, however, that binary compatibility is not guaranteed. See <http://www.synopsys.com/products/platforms> for the latest information.*

Accessing Memory Beyond 2 GB With 32-Bit Tools

In general, UNIX-based systems support a maximum memory of 2 GB for 32-bit processes. However, the VCS MX tool can extend memory beyond 2 GB.

Note:

Available memory is space not used by the OS, the windowing system, or other applications.

To access memory beyond 2 GB,

1. Make sure your server has Solaris 9 (or later) loaded.
2. Make sure your server has at least 4 GB of memory (physical and swap space) available.

Note:

Physical memory equals data size plus stack size, and stack size is used before data size. Therefore setting stack size to a large value causes problems for designs that need to go over 2 GB. If you set the stack size too high, you cannot get enough memory for your data. To check the settings, use the `limit` command at the system prompt. For more information, see “Configuring the Environment” in *Installing Synopsys Tools*.

3. Make sure the system you are using does not have restrictions that prevent you from using more than 2 GB of memory.
4. Create unlimited data size in the shell that you are using: C, Bourne, Korn, or Bash. If there are system-wide limits on the data size you can create, you can remove them or override them. You can do this in one of two ways:
 - Enter one of the following commands:

For the C shell,

```
% limit datasize 3800000
```

For the Bourne, Korn, or Bash shell,

```
# ulimit -s -d 3800000
```

- Modify the kernel of your server. This approach allows everyone using your server to extend memory beyond 2 GB.

Installing the Software

VCS MX uses the Synopsys Installer tool, which allows you to use a graphical user interface (GUI) or a text script. For information about downloading Synopsys Installer and VCS MX, see “Downloading the Software” in *Installing Synopsys Tools*

To install VCS MX by EST or from the CD, follow the procedures described in *Installing Synopsys Tools*

Example 1-1 in *Installing Synopsys Tools* shows a Synopsys media installation script for the synthesis tools. VCS MX is installed in a similar manner.

VCS MX is a stand-alone product and cannot be installed over an existing Synopsys product, including a prior version of VCS MX. You must create a new directory for VCS MX.

Setting the Environment Variables

This section discusses the following environment variables:

- VCS_HOME

- SNPSLMD_LICENSE_FILE

It is recommended that you place these variables in your \$HOME/.cshrc or \$HOME/.profile file as your default settings.

Setting the VCS_HOME Environment Variable

Follow these steps.

1. Set the VCS_HOME environment variable in the shell that you are using in which the *root_directory* argument is the name of the VCS MX root directory.
 - If you are using the C shell, enter

```
% setenv VCS_HOME /vcs_mx/build/vcs_mx/vcs_mxD-2009.12
```
 - If you are using the Bourne shell, enter

```
% VCS_HOME=/vcs_mx/build/vcs_mx/vcs_mxD-2009.12
export VCS_HOME
```
2. Add the directory containing the VCS MX executable files to the PATH environment variable.
 - If you are using the C shell, add the following line to the .cshrc file:

```
set path=($VCS_HOME/bin $path)
```
 - If you are using the Bourne, Korn, or Bash shell, add the following line to the .profile or .kshrc file:

```
PATH= (path:$VCS_HOME/bin)
export PATH
```

Setting the SNPSLMD_LICENSE_FILE Environment Variable

You must install the SCL software and define the SNPSLMD_LICENSE_FILE variable before you can verify the VCS MX installation.

For information about downloading and installing SCL, see Chapter 30, “Synopsys Common Licensing (version 10.9.2)” in *Installing Synopsys Tools*. For detailed instructions on setting the license variable, see “Setting Up the Licensing Environment” on page 30-8 in *Installing Synopsys Tools*.

Verifying the VCS MX Installation

To verify the VCS MX installation,

1. Make sure you are in a directory where you have read/write privileges.

```
% cp -r $VCS_HOME/doc/examples/basic-hdl/verilog/comp_run .  
% cd ./comp_run  
% vcs ./addr4.v -R -debug
```

If you see information about the product version, production date, and copyright, the installation was successful.

2. Run the DVE GUI on each installed platform by entering

```
% dve
```

3. Exit the GUI by choosing File > Exit in the DVE window.