

System Studio Installation Guide

Version E-2010.09

Sept. 2010

These installation notes present information about installing System Studio version E-2010.09 in the following sections:

- [Media Availability and Supported Platforms](#)
- [Memory Requirements](#)
- [Installing the Software](#)
- [Setting Up the User Environment](#)
- [Verifying the System Studio Installation](#)

Note:

The installation instructions in this chapter are the most up-to-date available at the time of production. However, changes might have occurred. For the latest installation information, see the product release notes or documentation.

See also <http://www.synopsys.com/Support/Licensing/Installation/Pages> for additional installation and licensing information.

Media Availability and Supported Platforms

System Studio is available by EST download upon initial software release, and at a later date on DVD. [Table 1](#) shows the supported compute platforms, operating systems, Synopsys platform keywords, and windowing environments for this release.

Table 1: Supported Platforms, Operating Systems, and Keywords

Compute Platform	Operating System	Synopsys Platform
x86_64, x86	Red Hat Enterprise Linux v4, 5	linux (32-bit / 64-bit mode)
x86_64, x86	SUSE Enterprise Linux v10, 11	linux (32-bit / 64-bit mode)
x86_64, x86	Windows XP, Server 2003	msvc50 (32-bit mode)

Memory Requirements

The System Studio tool has the following minimum memory requirements:

- Physical memory - 2GB (4GB recommended)

Installing the Software

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

Installation on Linux

System Studio uses the Synopsys Installer tool, which allows you to use a text script or a graphical user interface (GUI). For information about downloading the Synopsys Installer, see [Installing Synopsys Tools](http://www.synopsys.com/Support/Licensing/Installation/Pages) at <http://www.synopsys.com/Support/Licensing/Installation/Pages>. To install System Studio, follow the procedures described in [Installing Synopsys Tools](#). This document provides a Synopsys media installation script.

Installation on Windows

Before you can use System Studio on this platform, you need to set up the following environment. Please read the whole chapter, before you start the installation.

C/C++ and Fortran Compilers

Install Microsoft Visual Studio 2005 for C++, along with the following updates:

- Microsoft Visual Studio 2005 Professional Edition - ENU Service Pack 1 (KB926601)
- Update for Microsoft Visual Studio 2005 Professional Edition - ENU (KB932232)

To determine the C++ compiler and linker versions, type:

```
-> cl
Microsoft(R) 32-bit C/C++ Optimizing Compiler Version 14.00.50727.762 for 80x86

-> link /version
Microsoft(R) Incremental Linker Version 8.00.50727.762
```

Install Compaq Visual Fortran, if you are going to write your own Fortran models.

Cygwin

Install the latest version of Cygwin. You can download it from <http://www.cygwin.com/>. Click on the Install Now! button, and follow the instructions. Make sure that you install in addition to the default packages the following packages as an option:

- Devel/make
- Interpreters/perl
- Utils/time

Installing System Studio

Install System Studio inside the Cygwin shell to get the file and directory access modes right. Use a directory that has a short path without white-spaces. On the Windows platform System Studio does not need the Synopsys Installer tool. You just need to download the following two gzipped tar files:

```
css_vE-2010.09_common_msvc50.tar.gz
css_vE-2010.09_msvc50.tar.gz
```

from EST and unpack them from the Cygwin shell into the intended installation directory `<install_dir>`, e.g. `/u/tools/synopsys/CCSS/E-2010.09`. Use a directory that has a short path without white-spaces. You must not use WinZip or other Windows tools to unpack the archive! They change some files and links during the process and System Studio will not operate properly. Instead use the tar command from the Cygwin shell.

The following is an example for the installation process, that assumes that the downloaded tar files are located in the directory `<download_dir>`:

```
cd <install_dir>
gunzip -c <download_dir>/css_vE-2010.09_common_msvc50.tar.gz | tar xvf -
gunzip -c <download_dir>/css_vE-2010.09_msvc50.tar.gz | tar xvf -
```

Shell Environment Variable Settings

Please make sure that the compilers come first in your `PATH` variable! The two executables `/bin/df.exe` and `/bin/link.exe` in the Cygwin installation collide with the Fortran compiler and the linker of MS-Visual C/C++, respectively. If the Cygwin executables `df.exe` and `link.exe` come first in your `PATH` variable you will get an additional warning if you start System Studio. If you have not installed (and you do not need) the Fortran compiler the Cygwin executable `df.exe` will always be first in your `PATH`. In this case you should set the environment variable `CCSS_NO_FORTRAN` to 1 to avoid the warning. The `.bash_profile` or `.bashrc` file in `$HOME` contains variable settings for a bash shell.

Here is an example for your `.bash_profile`:

```
export CCSS_KEYS=<port>@<host>
export SNPS_ARCH=msvc50
export SYNOPSIS_CCSS="<drive>:/<path1>/$SNPS_ARCH/ccss"
export SYNOPSIS_CCSS_PATH=`cygpath $SYNOPSIS_CCSS`
export CCSS_NO_FORTRAN=1

export INCLUDE="C:\Program Files\Microsoft Visual Studio 8\VC\include;\
               C:\Program Files\Microsoft Visual Studio 8\VC\PlatformSDK\Include"
export LIB="C:\Program Files\Microsoft Visual Studio 8\VC\LIB;\
           C:\Program Files\Microsoft Visual Studio 8\VC\PlatformSDK\Lib;\
           C:\Program Files\Microsoft Visual Studio\DF98\LIB"

export PATH="$HOME/bin:$SYNOPSIS_CCSS_PATH/bin:\
            /cygdrive/c/Program Files/Microsoft Visual Studio 8/VC/bin:\
            /cygdrive/c/Program Files/Microsoft Visual Studio 8/Common7/IDE:\
            /cygdrive/c/Program Files/Microsoft Visual Studio/DF98/BIN:\
            /bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin"

export CCSS_SIM_DIR="<drive>:/<path2>"
```

Adapt the paths, if you installed the Microsoft compiler in a different place.

Please note that `SYNOPSIS_CCSS` starts with a drive letter and uses forward slashes as the directory separator. You may use UNC paths, too. But do not use a Cygwin path for this variable! Make sure that the strings of `<path1>` and `<path2>` are short and do not contain any spaces.

Don't use a Windows editor like notepad or wordpad to edit these files. They will use Windows line end markers, which will cause trouble in bash. Instead use an editor like `vi` or `emacs` that comes with Cygwin to edit them.

Starting System Studio

You have to start System Studio from a Cygwin bash shell with the command `ccss`. Starting with a double-click or from a DOS command line is not supported.

Acrobat Reader

Starting the online documentation from the help menu requires Adobe Reader 7.0 or later to be installed.

Setting Up the User Environment

[Table 2](#) lists the most important environment variables that you should set before starting the tool. A complete list of environment variables that you can set can be found in the System Studio Reference Manual.

Table 2: Most important environment variables

Variable name	Description
<code>SNPS_ARCH</code>	Identifies the platform (<code>linux</code> or <code>msvc50</code>)
<code>SYNOPSYS_CCSS</code>	Identifies the System Studio installation directory. For example: <code>\$SYNOPSYS/\$SNPS_ARCH/ccss</code>
<code>CCSS_KEYS</code> or <code>LM_LICENSE_FILE</code> or <code>SNPSLMD_LICENSE_FILE</code>	Port number and server name of the machine running your <code>snpslmd</code> license server.

You can choose different runtime configurations (compiler versions used to compile and link the simulations) in System Studio by setting the environment variable `CCSS_RUNTIME`. On Linux this should be done before sourcing the GNU tools setup file `$SYNOPSYS_CCSS/./gnupackages/source_me.[c]sh`. The possible runtime configurations for the different platforms are listed in [Table 3](#).

Table 3: Runtime configurations

Platform (<code>SNPS_ARCH</code>)	Runtime Configuration
<code>linux</code>	<code>gcc-3.3.6</code> , <code>gcc-4.1.1</code> , <i><code>gcc-4.2.4</code></i> , <code>gcc-4.4.3-64</code>
<code>msvc50</code>	<i><code>msvc-8.0</code></i> , <code>msvc-8.0-mt</code>

If `CCSS_RUNTIME` is not set, the default runtime configuration shown in *italic font* will be used. A typical setup on a Linux machine with a c-shell is shown below. It sets the above environment variables and includes the System Studio executable files into your executable search path. Also it configures System Studio to use the GNU compiler that

is shipped with System Studio by sourcing a setup file in the System Studio release. As an example for c-shell users you can add the following lines to your `.cshrc` setup file:

```
setenv SNPS_ARCH linux
setenv CCSS_RUNTIME gcc-4.4.3-64
setenv SYNOPSISYS_CCSS /ccss_home/$SNPS_ARCH/ccss
setenv CCSS_KEYS <port_nr>@<license_server>
set path = ($SYNOPSISYS_CCSS/bin $path)
source $SYNOPSISYS_CCSS/./gnupackages/source_me.csh
```

Setup Scripts for the GNU compiler and tools

System Studio comes with a package of GNU compilers and tools that you can setup by sourcing a script as shown in the example above. For c-shell users this is:

```
source $SYNOPSISYS_CCSS/./gnupackages/source_me.csh
```

and for bash users this is:

```
source $SYNOPSISYS_CCSS/./gnupackages/source_me.sh
```

Alternatively, if you are using a `Modules` system, you can use this system to do the correct setup. System Studio provides a `.tcl` script to simplify the setup and make it easier to maintain a `Modules` based setup. The following is a small example for a module file:

```
##Module1.0
# Set-up for System Studio

# basic setup

set snps_arch linux
setenv SNPS_ARCH $snps_arch
set ccss_runtime gcc-4.4.3-64
setenv CCSS_RUNTIME $ccss_runtime
set synopsisys_ccss /u/ccmaster/releases/ccss/2009.12-SP2-P1/$snps_arch/ccss
prepend-path PATH $synopsisys_ccss/bin
setenv CCSS_KEYS <port_nr>@<license_server>

# gnu-package setup

# source the tcl file containig the gnupackage setup procedures
source $synopsisys_ccss/./gnupackages/setup-gnupackages-module.tcl
# execute the setup for the gnupackage
setup-gnupackages-module $synopsisys_ccss
```

Setting up the Licensing

You must install the Synopsys Common Licensing (SCL) software and define the on of the variables `SNPSLMD_LICENSE_FILE`, or `CCSS_KEYS`, or `LM_LICENSE_FILE` before you can verify the System Studio installation. For information about downloading SCL, installing SCL, or setting the license variables, see [Installing Synopsys Tools at <http://www.synopsys.com/Support/Licensing/Installation/Pages>](http://www.synopsys.com/Support/Licensing/Installation/Pages).

Analyzing VHDL Files

If you intend to use the external simulation interface for VHDL cosimulation under the algorithmic domain of System Studio, you must ensure that the VHDL packages are analyzed before you use them. Your system administrator should analyze the files when System Studio is installed. The relevant commands are

```
> cd $SYNOPSYS_CCSS/packages/vsscli/src/  
> vhdlan -nc ccss_vsscli_package.vhdl  
> cd $SYNOPSYS_CCSS/packages/bittrue/vhdlsynopsys/src/  
> vhdlan -nc CCSS_PACKAGE_SYNOPSYS.vhdl LIB_0_0_1_PACKAGE_SYNOPSYS.vhdl
```

Verifying the System Studio Installation

To verify the System Studio Installation, make sure you are in a directory where you have read/write privileges, e.g.:

```
> cd $HOME
```

Invoke the Design Center by entering the following command on a licensed machine:

```
> ccss
```

If the System Studio Design Center appears on the screen the installation was successful.