Synopsys® Common Licensing Administration Guide

Version 2018.06, June 2018

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
690 E. Middlefield Road
Mountain View, CA 94043
www.synopsys.com

Contents

3. Before You Install Synopsys Common Licensing Software

4. Downloading SCL Software and Obtaining License Key Files

5. Configuring and Verifying the License Key File

Appendix A. Starting SCL on Windows Platforms With LMTOOLS

Appendix B. SCL Command-Line Interface

Appendix C. SCL License Key File Sort Utility

Appendix D. Custom Server Debug Log

Appendix E. SCL License Server in a Cloud Environment

Index

Preface

This preface includes the following sections:

- About This User Guide
- Customer Support

About This User Guide

This guide provides information about Synopsys Common Licensing (SCL) version 2018.06. It describes how to maintain the licensing software and provides troubleshooting guidelines.

Note:

> Install a single copy of SCL on your license server to be used with all Synopsys tools. Installation of Synopsys tools and SCL is not order dependent. That is, you can install SCL before or after you install your Synopsys tools. However, you cannot use any Synopsys tools reliant on SCL until you have installed and configured SCL.

Audience

The *Synopsys Common Licensing Administration Guide* is intended for the license network administrator responsible for installing and maintaining the SCL software. Client users of Synopsys tools who set their own licensing environment variables will also find this guide useful.

To use this guide, experience installing Synopsys tools is helpful, including setting environment variables for the client systems on which you run those tools. Familiarity with the Flexera Software FlexNet product makes it easier to install, configure, and administer SCL.

Related Publications

For additional information about the Synopsys Common Licensing software, see the documentation on the Synopsys SolvNet® online support site at the following address:

https://solvnet.synopsys.com/DocsOnWeb

You might also want to see the documentation for the following related Synopsys products:

- *Synopsys Installation Guide*

  This guide is available at http://www.synopsys.com/install

- *FlexNet Licensing End Users Guide*

  The *FlexNet Licensing End Users Guide* is available from Synopsys by clicking on the “LICENSING DOCS” link at http://www.synopsys.com/licensing. It is also delivered with the SCL software.

  SCL runs on various platforms. For information about supported platforms, operating systems, and keywords, see the *Synopsys Common Licensing Release Notes* on the SolvNet site or the Synopsys Licensing QuickStart Guide at

  http://www.synopsys.com/licensing.

- SolvNet articles pertaining to SCL

  See “Accessing SolvNet” on page xiii.

The *Synopsys Common Licensing Administration Guide* version 2018.06 is available in PDF format in the SCL installation directory at scl_root/doc/SCL_2018.06_Administration_Guide.pdf.

Release Notes

Information about new features, enhancements, changes, known limitations, and resolved Synopsys Technical Action Requests (STARs) is available in the *Synopsys Common Licensing Release Notes* on the SolvNet site.

To see the *Synopsys Common Licensing Release Notes*,

1. Go to the SolvNet Download Center located at the following address:

   https://solvnet.synopsys.com/DownloadCenter

2. Select Synopsys Common Licensing, and then select a release in the list that appears.

Conventions

The following conventions are used in Synopsys documentation.

Convention	Description
`Courier`	Indicates syntax, such as `write_file`.
`Courier italic`	Indicates a user-defined value in syntax, such as `write_file` *`design_list`*.
`Courier bold`	Indicates user input—text you type verbatim—in examples, such as `prompt>` **`write_file top`**
[ ]	Denotes optional arguments in syntax, such as `write_file [-format` *`fmt`*`]`
...	Indicates that arguments can be repeated as many times as needed, such as *`pin1 pin2 ... pinN`*
|	Indicates a choice among alternatives, such as `low | medium | high`
Ctrl+C	Indicates a keyboard combination, such as holding down the Ctrl key and pressing C.
\	Indicates a continuation of a command line.
/	Indicates levels of directory structure.
Edit > Copy	Indicates a path to a menu command, such as opening the Edit menu and choosing Copy.

Customer Support

Customer support is available through SolvNet online customer support and through contacting the Synopsys Technical Support Center.

Accessing SolvNet

The SolvNet site includes a knowledge base of technical articles and answers to frequently asked questions about Synopsys tools. The SolvNet site also gives you access to a wide range of Synopsys online services including software downloads, documentation, and technical support.

To access the SolvNet site, go to the following address:

https://solvnet.synopsys.com

If prompted, enter your user name and password. If you do not have a Synopsys user name and password, follow the instructions to sign up for an account.

If you need help using the SolvNet site, click HELP in the top-right menu bar.

Contacting the Synopsys Technical Support Center

If you have problems, questions, or suggestions, you can contact the Synopsys Technical Support Center in the following ways:

- Open a support case to your local support center online by signing in to the SolvNet site at https://solvnet.synopsys.com, clicking Support, and then clicking “Open A Support Case.”
- Send an e-mail message to your local support center.
  - E-mail support_center@synopsys.com from within North America.
  - Find other local support center e-mail addresses at

    https://www.synopsys.com/support/global-support-centers.html
- Telephone your local support center.
  - Call (800) 245-8005 from within North America.
  - Find other local support center telephone numbers at

    https://www.synopsys.com/support/global-support-centers.html

1

Introduction

This chapter introduces Synopsys Common Licensing (SCL). It gives an overview of SCL version 2018.06, which is based on FlexNet 11.14.1.3, the standard licensing software provided by Flexera Software.

In addition, this chapter describes the supported platforms and SCL features, including on-support licenses.

This chapter includes the following sections:

- Overview of Synopsys Common Licensing (SCL) Version 2018.06
- Summary of Tasks to Set Up SCL
- Supported Platforms
- Synopsys Legacy Vendor Daemons

Overview of Synopsys Common Licensing (SCL) Version 2018.06

Synopsys Common Licensing (SCL) provides a single method of obtaining licenses used by all Synopsys tools. SCL reduces licensing administration complexity, minimizing the effort you expend in installing, maintaining, and managing licensing software for Synopsys tools.

SCL version 2018.06 is based on the FlexNet (formerly FLEXlm) 11.14.1.3 release. SCL is based on combined vendor daemon technology from Flexera Software; this technology provides the capability to run a single vendor daemon, `snpslmd`, to serve licenses to all Synopsys products. The SCL version 2018.06 vendor daemon replaces all previously released Synopsys vendor daemons. For a complete list of legacy vendor daemons, see Synopsys Legacy Vendor Daemons.

The FlexNet `lmgrd` license daemon manages the initial interaction with the client application requesting a license. The `lmgrd` daemon also locates and starts the `snpslmd` daemon, passing to it the connection with the client application. The `snpslmd` daemon reads and interprets (parses) the license key file and then handles client license requests.

The Synopsys `snpslmd` vendor daemon maintains the usage counts for Synopsys licensed tools. In addition to standard FlexNet vendor daemon functionality, `snpslmd` also implements functionality that is specific to Synopsys to support features such as on-support licensing on the server.

Important:
You must use FlexNet version 11.14.1.3 or a later version of the `lmgrd` license daemon.

SCL version 2018.06 includes the following features.

- It uses the `SNPSLMD_LICENSE_FILE` or `LM_LICENSE_FILE` license environment variable.
- It is delivered as a standalone tool. That is, it is not included with the tools but available as a separate download.

In addition, SCL version 2018.06 provides on-support licensing. On-support licenses

- Are provided when you purchase maintenance for a product
- Allow for ongoing availability of the licensed feature as new versions of the feature become available
- Provide continuous support for a tool until the expiration date of the license. Generally, support continues for versions released up to two years after the version specified in the license key file

For details about on-support licenses, including an example of an on-support license key, see On-Support Keys.

Summary of Tasks to Set Up SCL

To use licensed Synopsys tools, you must

- Obtain and install SCL version 2018.06 software
- Download and configure the license key file for your site
- Start SCL by using the provided `lmgrd` license server daemon
- Set an environment variable on client systems to specify the location of the SCL license server

This guide covers these tasks in the chapters that follow, with the exception of SCL version 2018.06 installation. For instructions on installing SCL, see the *Synopsys Installation Guide* at http://www.synopsys.com/install.

Supported Platforms

SCL runs on various OS platforms. For information about supported platforms, operating systems, and keywords for the SCL server, see Table 1-1, Table 1-2, Table 1-3, and Table 1-4 or the *Synopsys Common Licensing Release Notes* on the SolvNet site.

Limitation:

On SUSE platforms, the license server might not start properly when the root file system is configured with the reiserfs file system type. The following error message displays in the license server log file.

```
22:36:30 (snpslmd) Cannot open daemon lock file
22:36:30 (snpslmd) EXITING DUE TO SIGNAL 41 Exit reason 9
22:36:30 (lmgrd) snpslmd exited with status 41 (Exited because another
 server was running)
22:36:30 (lmgrd) MULTIPLE "snpslmd" license server systems running.
22:36:30 (lmgrd) Please kill, and run lmreread
22:36:30 (lmgrd)
22:36:30 (lmgrd) This error probably results from either:
22:36:30 (lmgrd) 1. Another copy of the license server manager (lmgrd)
 is running.
22:36:30 (lmgrd) 2. A prior license server manager (lmgrd) was killed
with "kill -9"
22:36:30 (lmgrd) (which would leave the vendor daemon running).
22:36:30 (lmgrd) To correct this, do a "ps -ax | grep snpslmd"
22:36:30 (lmgrd) (or equivalent "ps" command)
22:36:30 (lmgrd) and kill the "snpslmd" process.
22:36:30 (lmgrd)
```

Solution: You must reimage your machine to use other SUSE-supported root file system types, such as ext3, ext4, or Btrfs.

To find the root file system on your machine, use the `df -print-type` command:

```
/var/tmp>  df -print-type
Filesystem Type 1K-blocks Used Available Use% Mounted on /dev/cciss/
c0d0p2
reiserfs 59438628 9059928 50378700 16% /
```

SCL is platform-independent. This means that you can run any combination of SCL clients and servers, providing the SCL server and client (tool) platforms are supported. Each client requires network access to a server through the Transmission Control Protocol (TCP).

For a list of server platforms supported by SCL, see Table 1-1, Table 1-2, Table 1-3, and Table 1-4.

Table 1-1 Supported Platforms, Operating Systems, and License Daemons in SCL Version 2018.06 Part 1

OS and hardware platform (Synopsys platform keyword)	**License daemon support**					
	sandwork	**pdld**	**synplctyd**	**vastlmd**	**cowared**	**synforad**
Red Hat Enterprise Linux x86 — 64-bit (linux64)	yes	yes	yes	yes	yes	yes
SUSE Linux Enterprise Server — 64-bit (suse64)	yes	yes	yes	yes	yes	yes
Microsoft Windows x86_64 — 32-bit (win32)	yes	yes	yes	yes	yes	yes

Table 1-2 Supported Platforms, Operating Systems, and License Daemons in SCL Version 2018.06 Part 2

OS and hardware platform (Synopsys platform keyword)	**License daemon support**					
	arcd/ mwflexd	**extremed**	**nsysnvs**	**ACAD**	**magma**	**riod**
Red Hat Enterprise Linux x86 — 64-bit (linux64)	yes	yes	yes	yes	yes	yes
SUSE Linux Enterprise Server — 64-bit (suse64)	yes	yes	yes	yes	yes	yes
Microsoft Windows x86_64 — 32-bit (win32)	yes	no	yes	yes	yes	yes

Table 1-3 Supported Platforms, Operating Systems, and License Daemons in SCL Version 2016.12 Part 3

OS and hardware platform (Synopsys platform keyword)	**License daemon support**				
	TAVEREN	**sclmgrd**	**snslmgrd**	**eved**	**knights**
Red Hat Enterprise Linux x86 — 64-bit (linux64)	yes	yes	yes	yes	yes

Table 1-3 Supported Platforms, Operating Systems, and License Daemons in SCL Version 2016.12 Part 3 (Continued)

OS and hardware platform (Synopsys platform keyword)	**License daemon support**				
	TAVEREN	**sclmgrd**	**snslmgrd**	**eved**	**knights**
SUSE Linux Enterprise Server — 64-bit (suse64)	yes	yes	yes	yes	yes
Microsoft Windows x86_64 — 32-bit (win32)	yes	yes	yes	yes	yes

Table 1-4 Supported Platforms, Operating Systems, and License Daemons in SCL Version 2018.06 Part 4

OS and hardware platform (Synopsys platform keyword)	**License daemon support**						
	perflmd	**targetld**	**SIINT**	**atrenta**	**simple**	**wlld**	**All other Synopsys license daemons[1]**
Red Hat Enterprise Linux x86 — 64-bit (linux64)	yes	yes	yes	yes	yes	yes	yes
SUSE Linux Enterprise Server — 64-bit (suse64)	yes	yes	yes	yes	yes	yes	yes

Table 1-4 Supported Platforms, Operating Systems, and License Daemons in SCL Version 2018.06 Part 4 (Continued)

OS and hardware platform (Synopsys platform keyword)	**License daemon support**						
	perflmd	**targetId**	**SIINT**	**atrenta**	**simple**	**wlld**	**All other Synopsys license daemons[1]**
Microsoft Windows x86_64 — 32-bit (win32)	yes	yes	yes	yes	yes	yes	yes

[1] All other Synopsys license daemons: adalmd, anagram, avantd, chrysalisd, hscd, metasoftd, saber_dmn, tmald, innologd, ISE-TCADd, EPIC, ssilmd, synopsysd, vcsd, nassd, numeritchd, TE_CATS, CADABRA, snpsOEM1, and sigmacd.

Important:

- The Sentinel SuperPro dongle is not supported on Windows platforms for server-based licenses. It is supported only for uncounted (serverless) licenses on Windows.
- The RSoft dongle uses the “RSOFT=” prefix. The RSoft dongle is supported on the Linux and Windows OS platforms.

Synopsys Legacy Vendor Daemons

Table 1-1 shows the legacy vendor daemons incorporated into SCL version 2018.06. (Other vendor daemons are not supported. Tools based on such daemons are not compatible with SCL version 2018.06.) Before starting SCL, these daemon processes must be shut down. See Using the lmgrd Command to Start SCL.

Table 1-5 Legacy Daemons Incorporated Into SCL Version 2018.06

Acquired Company	Daemons
Analog Design Automation	`adalmd`
Atrenta Inc.	`atrenta`
Avant! Corporation	`anagram`, `avantd`, `chrysalisd`, `hscd`, `metasoftd`, `saber_dmn`, `tmald`
CHIPit	`pdld`
CoWare, Inc.	`cowared`
EVE	`eved`
Extreme DA	`extremed`
Hitachi High-Tech	`SIINT`
Innologic Systems, Inc.	`innologd`
Integrated Systems Engineering	`ISE-TCADd`
Legacy and other daemons	`EPIC`, `ssilmd`, `synopsysd`, `vcsd`
Magma Design Automation	`ACAD`, `magma`, `riod`, `TAVEREN`, `knights`
Nassda Corporation	`nassd`
nSys Design Systems	`nsysnvs`
Numerical Technologies	`numeritchd`, `TE_CATS`, `CADABRA`
PerfectVIPs	`perflmd`
Placeholder for future OEM partners	`snpsOEM2`, `snpsOEM3`
Sandwork Design	`sandwork`

Table 1-5 Legacy Daemons Incorporated Into SCL Version 2018.06 (Continued)

Acquired Company	**Daemons**
SIGMA-C	`sigmacd`
Simpleware	`simple`
SpringSoft	`sclmgrd`, `snslmgrd`
Synfora	`synforad`
Synplicity	`synplctyd`
Target Compiler Technologies	`targetld`
VaST Systems Technology Corporation	`vastlmd`
Virage Logic	`arcd`, `mwflexd`
WinterLogic	`wlld`
Zenpire Corporation	`snpsOEM1`

2

Migrating to Synopsys Common Licensing

This chapter describes the steps that are required to migrate to Synopsys Common Licensing (SCL) version 2018.06. It includes the following sections:

- Planning Your Migration
- Hardware, Operating System, and Other Requirements

Planning Your Migration

You should maintain multiple single license servers to achieve fault tolerance. For example, you might split 4,000 license seats among two separate license servers, each running a separate license file. You must consider license pooling and rollover functionality when creating your server migration plans. Make sure your license server hardware meets SCL requirements. See Supported Platforms.

License Pooling Functionality

License pooling is the ability of an application to check out licenses from two or more servers at the same time and should not be confused with wide area network (WAN) functionality. Each server in the pool must be running the same version of SCL.

Note:

> Synopsys tools support license pooling but not license spanning. Synopsys tools supports the ability to check out one feature from server A and another feature from server B (pooling). But Synopsys tools do not support the ability to check out the same exact feature from two different servers at the same time (spanning).

Rollover Functionality

Rollover functionality allows the application to check out the license from the next SCL daemon in its search path (defined by the `SNPSLMD_LICENSE_FILE` or `LM_LICENSE_FILE` variable) if the current connected daemon is unavailable. This should not be confused with license WAN or pooling functionalities.

Applications do not roll over between the `snpslmd` combined vendor daemon and a legacy Synopsys license daemon if the current (connected) daemon is unavailable, even if both servers are in the license search path.

Note:

> After you have migrated to the `snpslmd` combined vendor daemon on all of your servers, this problem does not occur.

Hardware, Operating System, and Other Requirements

The following requirements ensures that your migration to the SCL `snpslmd` vendor daemon is successful.

- SCL Server Platform Requirements
- Hosting a Server With Other Vendor Daemons

- Single Server Configuration
- File Descriptor Limit
- TCP and Network Settings
- WAN License Servers
- Compute Farm and License Server
- Network Mounted File System
- Options File Use Guidelines

SCL Server Platform Requirements

The best practice is to use a supported version of the Red Hat Enterprise Linux (RHEL) version 7.x operating system, especially for WAN servers (CentOS can also be used since it is binary-compatible with RHEL). Using other supported server platforms is not recommended. The SCL snpslmd daemon supports both AMD and Intel x86_64 processors. The following SCL license server setup is recommended:

- A physical machine (virtual machines degrade the server performance).

  Note:

  Containers, such as Docker or Linux containers, are not supported.
- AMD or Intel x86_64 hardware running a supported version of Red Hat Enterprise Linux (RHEL) version 7.3.
- A CPU speed of 2 GHz or better.
- Multiple cores (This allows the `lmgrd` and the `snpslmd` processes to run on separate CPUs.)
- 8 GB of memory.
- 2 GBps Ethernet card and 2 GBps or better network switch.

If you are serving large license quantities (more than 300 socket connections to the server), make sure that you meet these requirements. In addition, do not host more than 4,000 licenses on a single license server or you might see license failures due to the load on the server machine.

Note:

Certain other OS platforms are supported, but they are recommended only if the number of licenses is small. For a list of supported platforms, see Table 1-1, Table 1-2, Table 1-3, and Table 1-4 or the *Synopsys Common Licensing Release Notes* on the SolvNet site.

http://www.synopsys.com/Support/LI/Licensing/Pages/SupportedOS.aspx

If you do not meet these requirements or plan to host the license server on any other OS platforms, contact the Synopsys Support Center at

https://solvnet.synopsys.com/EnterACall

Hosting a Server With Other Vendor Daemons

It is recommend that you do not host a Synopsys license daemon on the same machine with other license daemons based on FlexNet. If you must share the same machine with other daemons or processes, make sure that your license server meets the specifications for the hardware platform, including multiple cores of 2 GHz or better, 8 GB memory, and a 2 GBps Ethernet card. See SCL Server Platform Requirements.

Single Server Configuration

A single license server configuration is strongly recommended over a redundant, triad server configuration. The SCL components are most robust when they are part of a single-server configuration. The redundant server configuration adds an additional load on the license servers, which can increase latency, eliminating any advantage of redundancy.

Rather than using a redundant server configuration, it is recommended that you host multiple single-server configurations. This ensures that if the connection to one of the daemons is broken, some licenses will still be available; whereas if a redundant configuration fails (loss of quorum), no licenses will be available.

Important:

> If you are serving a large quantity of licenses, make sure you do not allow more than 4,000 socket connections to a single SCL license server. (In general, a license checkout consumes one socket connection.) If you go over this limit, you might experience degradation in performance or license failures due to the load on the server. In this case, it is recommended that you divide licenses among multiple single license servers.

File Descriptor Limit

Depending on the TCP/IP setting on your machine, it can take from several minutes to over two hours for the server machine to release the socket resources. To ensure that the file descriptor resources are not consumed, set the file descriptor to eight times the actual quantity of licenses served by the license daemon. For example, if your license file has 1,024 licenses, set the soft limit of the file descriptor to 8192. This change ensures that enough sockets are available.

TCP and Network Settings

Do not change any TCP settings on the server machine. If you have made any changes to the default settings, the license server might not operate properly.

Make sure the network card is configured for full duplex communication. In addition, make sure the network switch is configured for 1 GBps data transmission.

WAN License Servers

Synopsys provides the optional capability to host on a WAN (this applies to situations where the distance between the client and server is greater than 5 miles). You must have a contractual agreement with Synopsys to host a license server on your global network. If you need this capability, but are not contracted for it, contact your sales representative.

If you are planning to host a license server on a WAN, make sure that you meet the following requirements:

- For a WAN server, you should never run Synopsys Common Licensing on a virtual machine. Virtual machines degrade performance and increase latency.
- A single-server configuration is mandatory if you are setting the license server over a WAN. A redundant license server configuration increases the license checkout delays for the application.
- If the network latency is more than 30 milliseconds between the license server machine and the client machine, performance degradation occurs in the application startup. It is recommended that the license administrator evaluate whether the performance degradation due to network latency is acceptable to end users before setting up the WAN servers.

  Use the `ping` command to see the network latency. The usage in the following example shows that you are logged on to the client and elvis is the server:

  ```
  % ping elvis
  PING elvis (10.15.43.201) 56(84) bytes of data.
  64 bytes from elvis (10.15.44.206): icmp_seq=0 ttl=253 time=0.706 ms
  64 bytes from elvis (10.15.44.206): icmp_seq=1 ttl=253 time=0.267 ms
  64 bytes from elvis (10.15.44.206): icmp_seq=2 ttl=253 time=0.293 ms
  64 bytes from elvis (10.15.44.206): icmp_seq=3 ttl=253 time=0.241 ms
  64 bytes from elvis (10.15.44.206): icmp_seq=4 ttl=253 time=0.455 ms
  ```

  This example shows the usage of the `ping` command on the linux64 platform using the Red Hat Enterprise Linux operating system. For other operating systems and platforms, check the man pages for the `ping` command to determine the appropriate use to determine the network latency between the server and the client.

- Some applications check out more than one license, which might cause the tool to take longer to complete the job. Evaluate all the tools in your WAN to find out whether the license checkout delays are acceptable to the end user.
- Make sure that you have no more than two license servers in your license server search path. In addition, evaluate the delay while pointing to multiple license servers.

Set up the WAN servers only after evaluating these requirements.

Compute Farm and License Server

If you are using a compute farm, make sure that the license server and the compute farm are located within the same subnet. In addition, make sure that the license server meets the hardware and OS requirements as recommended by Synopsys.

Network Mounted File System

Install SCL on a local disk. Also, save the license key file, options file, and report log on the local disk. If these files are saved on a network disk, any glitches to the network file system affects the license server.

Options File Use Guidelines

The TIMEOUT or TIMEOUTALL options file keyword has a minimum value of 3600 seconds.

Add the AUTOMATIC_REREAD OFF keyword to the options file to prevent FlexNet from automatically rereading the license file.

Never invoke the `lmreread` command on a license server to read the changes made to the options file. Instead, restart the license server if you want the changes from the options file to be read properly by the license server.

All option file keywords must be specified in uppercase letters, such as RESERVE, MAX, INCLUDE, EXCLUDE, and so on. Keywords specified in lowercase letters are not recognized.

You cannot use both RESERVE and MAX for the same feature name.

If you have PACKAGE lines in the license key file, do not specify any license checkout policies such as RESERVE, INCLUDE, or EXCLUDE in the options file for any package or its components. Packages already have a license checkout policy, and specifying any license checkout policies might conflict with the policy specified in the options file.

3

Before You Install Synopsys Common Licensing Software

This chapter describes tasks you perform before installing and configuring Synopsys Common Licensing (SCL). Before you install the SCL software the first time, you need to complete most of the tasks described in this chapter. Whenever you change your licensing network, you must repeat these tasks. This chapter explains the `sssverify` utility and common licensing network configurations.

Information in this chapter applies to SCL on Linux and Windows platforms.

This chapter includes the following sections:

- Checking the Integrity of the License File
- License Server Configurations

Checking the Integrity of the License File

When you receive a new Synopsys license key file, it is mandatory that you run the `sssverify` utility (`sssverify` replaces `lmcksum`). The `sssverify` utility validates the integrity of the license file and detects any formatting errors in the file. You must also use this utility whenever the license file is updated.

The following topics are included in this section:

- Using the sssverify Utility
- Mapping Features to Products

Using the sssverify Utility

You must use the complete SCL license file received from Synopsys. Do not edit or merge the SCL license file with other license files.

Important:

> Synopsys license files in the combined vendor daemon format include security keys that the `snpslmd` vendor daemon uses to validate the authenticity of your license file and verify that keys have not been added or subtracted. Temporary license keys include the `SSST` feature in the key file. Monthly Technology Subscription Licenses (also known as peak licenses) and other types of revenue keys include the `SSS` feature.

Before starting the license server, you should verify the license key file by running the `sssverify` utility by entering the following code:

```
% cd scl_root/platform/bin
% sssverify  /path/to/synopsys.lic
```

For additional information about `sssverify`, see Using sssverify to Troubleshoot a Synopsys License File.

Mapping Features to Products

The `sssverify` utility can display all features associated with a product. The information displayed is comprehensive if the license file has an unmodified product comment section as received from Synopsys.

Use the following syntax to run the `sssverify` utility:

```
sssverify license_file [-pinfo | -ptof product_name |
   -ftop feature_name] [-o <output-file>] [-help]
```

Option	Description
license_file	Your Synopsys license file.
-pinfo	Displays all the product names and the associated features in the license file.
-ptof *product_name*	Displays all the features associated with a product.
-ftop *feature_name*	Displays the product and all associated features.
-o *output_file*	Redirects the product feature information to a file.
-help	Displays help information.

Note:

> The `-ptof` option does not work if product comment lines are removed. The `-pinfo` option works but does not display product names.

License Server Configurations

This section describes the following common configurations for licensing systems:

- Single Server
- Multiple Single License Servers
- Three-Server Redundancy
- Recommended License Server Configuration
- License Environment Variables
- Standard FlexNet License File List Behavior

For licensing systems that use more than one server, for each server you must take into account the considerations and tasks described in this section.

Single Server

You can configure your license system to use a single server to which all clients refer for their license requests. In this case, you set the `SNPSLMD_LICENSE_FILE` or `LM_LICENSE_FILE` environment variable on each client node to refer to the single server. The `SNPSLMD_LICENSE_FILE` environment variable is preferred but legacy software versions sometimes require the use of the `LM_LICENSE_FILE` environment variable.

Multiple Single License Servers

You can configure a client to refer to multiple single (distributed) license servers with each server serving a unique subset of all your licenses. In this way, you can distribute all your licenses across multiple servers.

Three-Server Redundancy

Three-server redundancy is a license server configuration in which the same license key file and vendor daemon is installed on three separate license servers so as to provide continued operation in the case of a single server failure. This provides fail-over protection only. When at least two of the three license servers are running and communicating, the system serves licenses as needed.

In this configuration, all three license servers operate to form a triad. The license servers send periodic messages to each other to make sure that at least two servers are running and communicating. A quorum is formed when at least two of the three license servers are running and communicating with each other.

If you are running license servers configured for three-server redundancy, you need to maintain an identical copy of the license file, as well as the `lmgrd` and the vendor daemons binaries, locally on each system rather than on a file server. If you do not do this, you lose all of the advantages of having redundant servers since the file server holding these files can become a single point of failure.

Warning:

> The use of redundant servers over a WAN is not supported. For a local area network (LAN), use of redundant servers is supported but not recommended. This method is not very robust. Poor network communication can cause system performance to slow. Slow network communication can also cause a delay in the transmission of heartbeats between license servers.

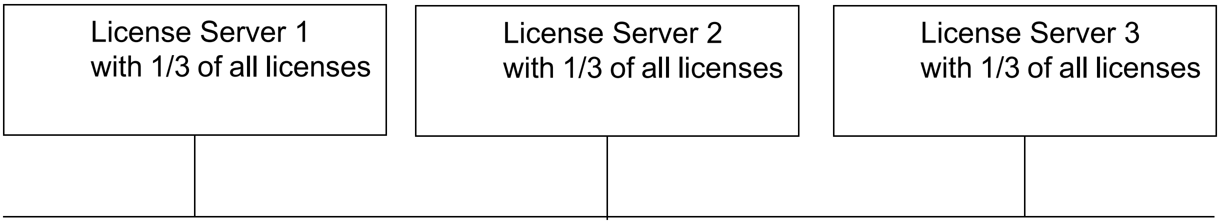
In addition, do not use the `lmreread` utility on large license files or on redundant servers. This utility is unreliable and can put the server in an unpredictable state. Sometimes on large license files, it might take several minutes to reread the license file. During this time, license denials are issued.

Recommended License Server Configuration

A single license server configuration is strongly recommended, as opposed to a redundant server configuration. The redundant server configuration adds an additional load on the license server and can hinder seamless operation of the license server, eliminating any advantage of redundancy. The Flexera Software FlexNet components are most robust when the daemons are in a single-server configuration.

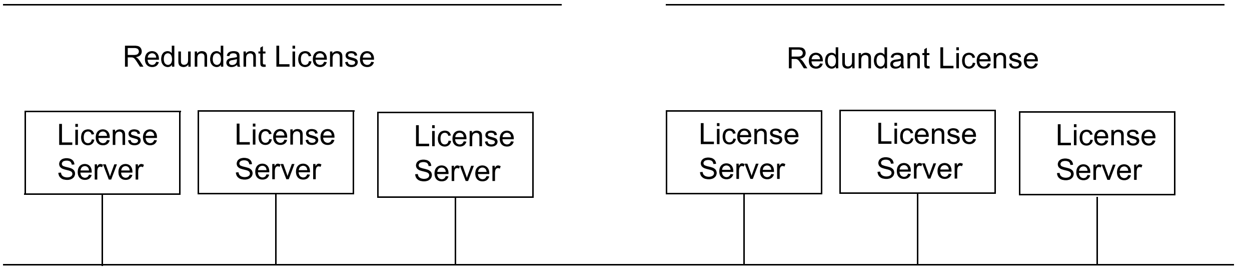
For redundancy, hosting multiple single-server configurations is recommended. This way, if one of the daemons goes down, only some of the licenses is unavailable instead of all the licenses in the case of the failure of a redundant configuration (loss of quorum).

The following example shows a single license server configuration:



The total number of licenses has been split equally amongst all three single license servers.

The following example shows a redundant license server configuration:



If you use WAN license servers, make sure that you use at least one local license server along with the WAN license servers. Also, make sure that your license environment variable points first to the local license server and then to the WAN license servers. For additional information about license environment variables see License Environment Variables.

License Server Configurations Summary

You can use the information provided in Table 3-1 to help you determine the optimal license server setup for your site.

Table 3-1 License Server Configuration Summary

Single server	Multiple single servers (distributed)	Three-server redundancy
Advantages		
Easy administration and monitoring - Single license file - Single server Simple to understand Single point for all licensing activity	Licensing distributed among multiple servers according to use patterns No single point of failure Some licenses almost always available Easy to understand	Fault tolerant - No single point of failure - All licenses are available if two servers are up (quorum) One license file for all licenses Automatic rollover
Disadvantages		
Single point of failure Not scalable	Server loss = Some licenses not available Complex maintenance	Quorum loss = No licenses Requires a robust network; redundant servers could introduce problems Each server must be maintained in its own environment
Recommended use		
Small to medium sites Sites where a central server controls all data	Geographically dispersed medium to large sites Sites that require fault tolerance	Not recommended

Note:

Use the information in Table 3-1 as a general reference only, taking into account special requirements for your site.

License Environment Variables

You set license environment variables on every client to identify the location of the SCL license server to be used. SCL supports the FlexNet vendor-specific license environment variable, `SNPSLMD_LICENSE_FILE`, and the standard FlexNet `LM_LICENSE_FILE` variable.

To locate the SCL license server to be used, each client must have one of its license environment variables set to point to the software. The license environment variables refer to one or more license servers or license key files to be used.

SCL uses the following two license environment variables, common to all Synopsys tools:

- `SNPSLMD_LICENSE_FILE`

  This is the Synopsys specific (vendor) license environment variable. You should use the `SNPSLMD_LICENSE_FILE` environment variable because the latest Synopsys tools read this environment variable first. License requests can be handled faster if the Synopsys tools do not have to contact servers from other vendors.

  You can assign multiple values to the `SNPSLMD_LICENSE_FILE` variable, but all of these values should refer to SCL license servers and not to other vendor products. If you set this variable to multiple values, the license key files on the servers listed are searched from left to right.

- `LM_LICENSE_FILE`

  The `LM_LICENSE_FILE` license environment variable, which is the FlexNet default, can be used to refer to multiple license key files, including those provided by vendors other than Synopsys. If multiple file locations are assigned to this variable, the Synopsys tools search those files from left to right to find the requested license.

Important:

> To speed tool startup, most Synopsys tools do not look at the `LM_LICENSE_FILE` variable if the `SNPSLMD_LICENSE_FILE` variable is set.

Specifying Values for License Environment Variables

To enable the use of multiple single license servers, you assign a list of servers to the `SNPSLMD_LICENSE_FILE` or `LM_LICENSE_FILE` license environment variable on the client node. Legacy software versions sometimes require the use of the `LM_LICENSE_FILE` environment variable.

For all SCL clients, use the *port@host* method to identify the license server to be used. Set the `LM_LICENSE_FILE` or `SNPSLMD_LICENSE_FILE` environment variable to port@host, by using the port and host values specified on the SERVER line in the license key file.

The following example illustrates how to specify the variable for a single license server:

```
% setenv SNPSLMD_LICENSE_FILE 27020@server1
```

or

```
% setenv LM_LICENSE_FILE 27020@server1
```

To specify multiple single license servers, use a colon (:) as the delimiter. The following example illustrates this specification for a Linux system:

```
% setenv SNPSLMD_LICENSE_FILE 27020@server1:26585@server2
```

or

```
% setenv LM_LICENSE_FILE 27020@server1:26585@server2
```

For Windows operating systems, use a semicolon (;) rather than a colon (:) as the delimiter.

If you are using a combination of local and WAN license servers, make sure that the `SNPSLMD_LICENSE_FILE` or `LM_LICENSE_FILE` environment variable points first to the local license servers and then to the WAN license servers.

The following example illustrates this specification for a Linux system:

```
% setenv SNPSLMD_LICENSE_FILE 27020@local_server1:26585@wan_server2
```

or

```
% setenv LM_LICENSE_FILE 27020@local_server1:26585@wan_server2
```

For a redundant server configuration, you can specify the `SNPSLMD_LICENSE_FILE` or `LM_LICENSE_FILE` environment variable using a comma separated list of the three port@host specifiers to identify the three license servers. The following example illustrates this specification for a Linux system:

```
% setenv SNPSLMD_LICENSE_FILE 27020@server1,27020@server2,27020@server3
```

or

```
% setenv LM_LICENSE_FILE 27020@server1,27020@server2,27020@server3
```

For additional information about setting the license environment variable on Linux and Windows systems, see Describing the License Environment Variables.

Selecting Host Nodes to Be Used As Servers

Whether you use a single server, distributed servers, or three-server redundancy, you need to identify the nodes to use as the license server host and communicate the host ID information about the server nodes to Synopsys. You can determine the host ID by using the `lmhostid` command.

Depending on your hardware and system configuration, you might see degradation in response times when the server goes above 4,000 simultaneous connections. In this case, it is recommended that you divide features among license servers.

The rotate_lic_server.sh Script

Synopsys provides you with an example script named rotate_lic_server.sh that you can use to randomly rearrange the order of license servers. This script assists in distributing the licenses evenly on to the different license servers. The script is included in the scl_root/ examples directory. It uses a pseudo round-robin scheduling method to randomly rearrange the order of license servers.

You can use either the servers specified in the `LM_LICENSE_FILE` variable from your environment or use the default servers provided in the script. The default setting is: ${LICENSE_PATH}

Usage:

```
% rotate_lic_server.sh [-e] [-s] [-l lic_path] [cmd args]
```

Here are the available options and arguments:

- The `-e` option uses the servers specified in the `LM_LICENSE_FILE` variable from your environment.
- The `-s` option uses the servers specified in the `SNPSLMD_LICENSE_FILE` variable from your environment.
- The `-l` option uses the specified license server path as the license server list.
- The `cmd` argument specifies the Synopsys application command.
- The `args` argument specifies the arguments passed to the Synopsys application command.

If the `cmd` argument is not provided, the script simply echoes the reordered license list. This option can be used in startup scripts, such as in the .login script as shown in the following example:

```
export LM_LICENSE_FILE=`rotate_lic_server.sh -l license_path`
```

You can customize the path to the license servers at the top of the script. For example,

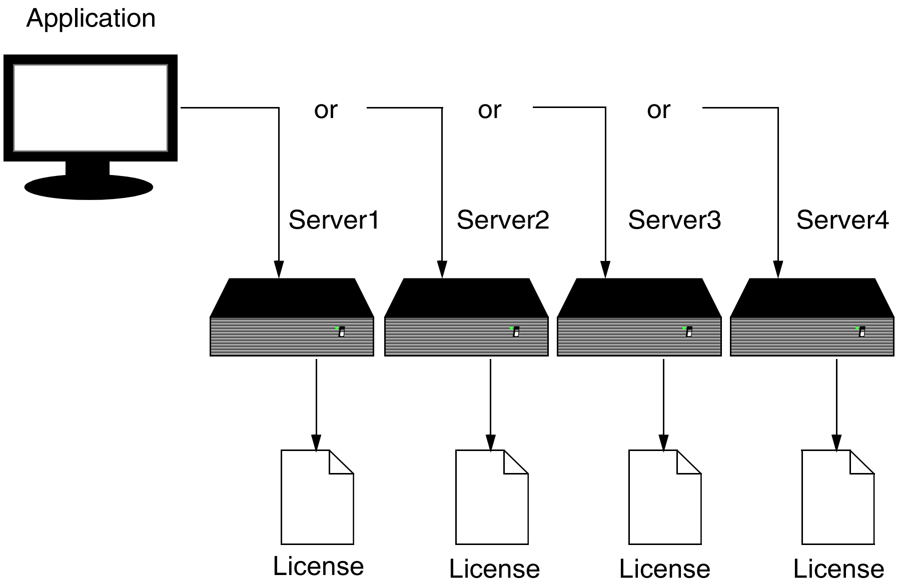
```
# Customize your license server here.
LICENSE_PATH="26585@host1:26585@host2:26585@triad1,26585@triad2,26585@tri
ad3"
```

Standard FlexNet License File List Behavior

Standard FlexNet license file list behavior is implemented in all Synopsys tools. For standard license file list behavior, servers are queried in the order listed in the license file list as specified in the `SNPSLMD_LICENSE_FILE` or `LM_LICENSE_FILE` environment variable settings. If the requested license is not available on the first server, the next one is tried, and so on.

Figure 3-1 illustrates standard FlexNet server selection behavior. For this example, assume that these four servers are listed in the following order in the license file list: Server1, Server2, Server3, and Server4. If Server1 can satisfy the license request, that server is used for all subsequent requests. If it cannot, then the next server specified in the license file list, Server2, is tried, and so on, until one of the servers specified in the list satisfies the license. Figure 3-1 shows this scenario.

Figure 3-1 Standard FlexNet License File List Server Selection Behavior



Rollover Behavior When Server Goes Offline

Following standard FlexNet behavior, if the Synopsys tool loses its connection to the currently used server, it attempts to locate a replacement server. The tool locates and connects to the first server identified in the license file list (specified by `SNPSLMD_LICENSE_FILE` or `LM_LICENSE_FILE`) that offers the required license.

The client application is terminated only if no other server specified in the license file list can satisfy the license requirement. For additional information about rollover behavior, see Rollover Functionality.

4

Downloading SCL Software and Obtaining License Key Files

This chapter includes the following sections:

- Registering for SolvNet and Downloading SCL
- Obtaining the Synopsys License Key File

Registering for SolvNet and Downloading SCL

To download Synopsys Common Licensing (or any Synopsys software) by Electronic Software Transfer, or to obtain license keys from the SmartKeys page, you must be registered to access the SolvNet site.

To register for access to the SolvNet site, go to the SolvNet Web page at https://solvnet.synopsys.com and click Sign up now. (This site also has information about how to retrieve a lost user name or password.)

Note:

To register for access to the SolvNet site, you must have a corporate e-mail address and know your Synopsys site ID. If you do not know your site ID, obtain it from your site administrator or from your sales representative.

To download the SCL software, obtain the download instructions for SCL 2018.06 from the SolvNet Download Center at https://solvnet.synopsys.com/DownloadCenter.

Obtaining the Synopsys License Key File

This section describes how to obtain a soft copy of your license key file from Synopsys by using the SmartKeys delivery system. It includes the following sections:

- About the License Key File
- Logging In to SmartKeys
- Retrieving Term Licenses
- Retrieving Monthly Technology Subscription Licenses

About the License Key File

Licensing data required for SCL to grant licenses is stored in a text file called the license key file. Synopsys creates the license key file containing all the information required to grant (or deny) license requests based on your purchased products and license availability.

When a new order is placed, Synopsys sends the license key file by e-mail to the registered site contact. Alternatively, anyone who is registered can retrieve the keys.

See Explaining the License Key File for an example license key file and details about how to interpret the license key file contents.

Logging In to SmartKeys

You can access SmartKeys through the SolvNet site (click Downloads, then click SmartKeys) or directly from the SmartKeys Web page at

https://solvnet.synopsys.com/SmartKeys

If requested, enter your SolvNet user name and password.

Note:

> You can retrieve keys for registered sites only. To determine the sites you are registered for or to add a site, log on to the SolvNet site at https://solvnet.synopsys.com and click My Profile at the top of the page.

On the SmartKeys Web page, click Key Retrieval to retrieve your latest technology subscription licenses, or click Monthly TSL Redemption to generate and retrieve any available monthly technology subscription licenses.

Technology subscription licenses are long-term keys that are regular licenses defined by an explicit contract period, such as one, three, or five years. Typically, these keys are issued for one year at a time. When the keys expire, they are extended for another year until the contract period expires.

Monthly technology subscription licenses (peak keys) are short-term licenses that are prepurchased to handle peak usage situations. These licenses can be redeemed for any 30-day range within the redemption period, which is usually a year. A monthly technology subscription license expires at the end of the redemption period—even if it has not been redeemed.

Retrieving Term Licenses

From the SmartKeys Web page,

1. Click Key Retrieval.
2. Enter the following information in the text boxes of the Key Retrieval dialog box:
   - Site ID (required)

     This is the Synopsys site ID number of the site for which you want to obtain term license keys. You can retrieve keys only for sites for which you are registered. Also, you can retrieve keys only for one site ID at a time. (For additional sites, make a new request.)
   - Attachment option – The default is ASCII text format.

     Do not change this option unless you are unable to receive text file (.txt) attachments by e-mail.

3. Click Retrieve Licenses to submit your request. The license key files for the chosen site are e-mailed to your SolvNet-registered e-mail address.

Retrieving Monthly Technology Subscription Licenses

From the SmartKeys Web page,

1. Click Monthly TSL Key Redemption.
2. Enter your site ID in the Site ID dialog box.
3. On the License Redemption Selection page, check the box next to the “Sales doc” number that you need.

   Note:

   This line also contains other descriptive information to assist you in making the right choice. For example, information about the Synopsys material number, material description, remaining quantity, and redemption period are included.

4. On the Monthly TSL Key Redemption License Details page, enter the necessary information in the appropriate text box:

   ❍ Withdrawal QTY – Enter the number of keys needed, up to the quantity available.

   ❍ Start Date – Enter the start date.

   ❍ Host ID 1 – Enter the host ID for the license server. The host ID must match the results of the `lmhostid` command (the `lmhostid` utility is included with SCL), and the license server must be running a supported hardware platform and operating system. For detailed platform information, go to http://www.synopsys.com/Support/LI/SupportPlatform/ReleaseSupport.

   ❍ Host ID 2 (Optional) – For redundant servers only (not recommended). Enter the host ID for the second license server (of three).

   ❍ Host ID 3 (Optional) – For redundant servers only (not recommended). Enter the host ID for the third license server (of three).

   Note:

   It is strongly recommended that your monthly technology subscription license keys be hosted on a different license server than your production (long term) licenses. This ensures that the production server is not affected.

5. Click Next to continue.
6. Click Generate to have the keys sent to the registered SolvNet e-mail address.

5

Configuring and Verifying the License Key File

This chapter provides detailed information about configuring and verifying the license key file on Linux and Windows systems. The following sections are included:

- Viewing the SCL Licensing Installation Directory Structure
- Verifying the License Key File
- Required License Key File Modifications
- Locating and Using the Options File

Note:

For detailed information about using SCL on Windows platforms, see Starting SCL on Windows Platforms With LMTOOLS.

Viewing the SCL Licensing Installation Directory Structure

The installation program installs the software into a directory structure created by the SCL installation program as shown in Figure 5-1. This directory structure is the same for all platforms. Only the platform directory name changes, depending on the platform you are installing. For example, *platform* might be “linux64” or “win32”.

Figure 5-1 Installation Directory Structure and File Location and Description

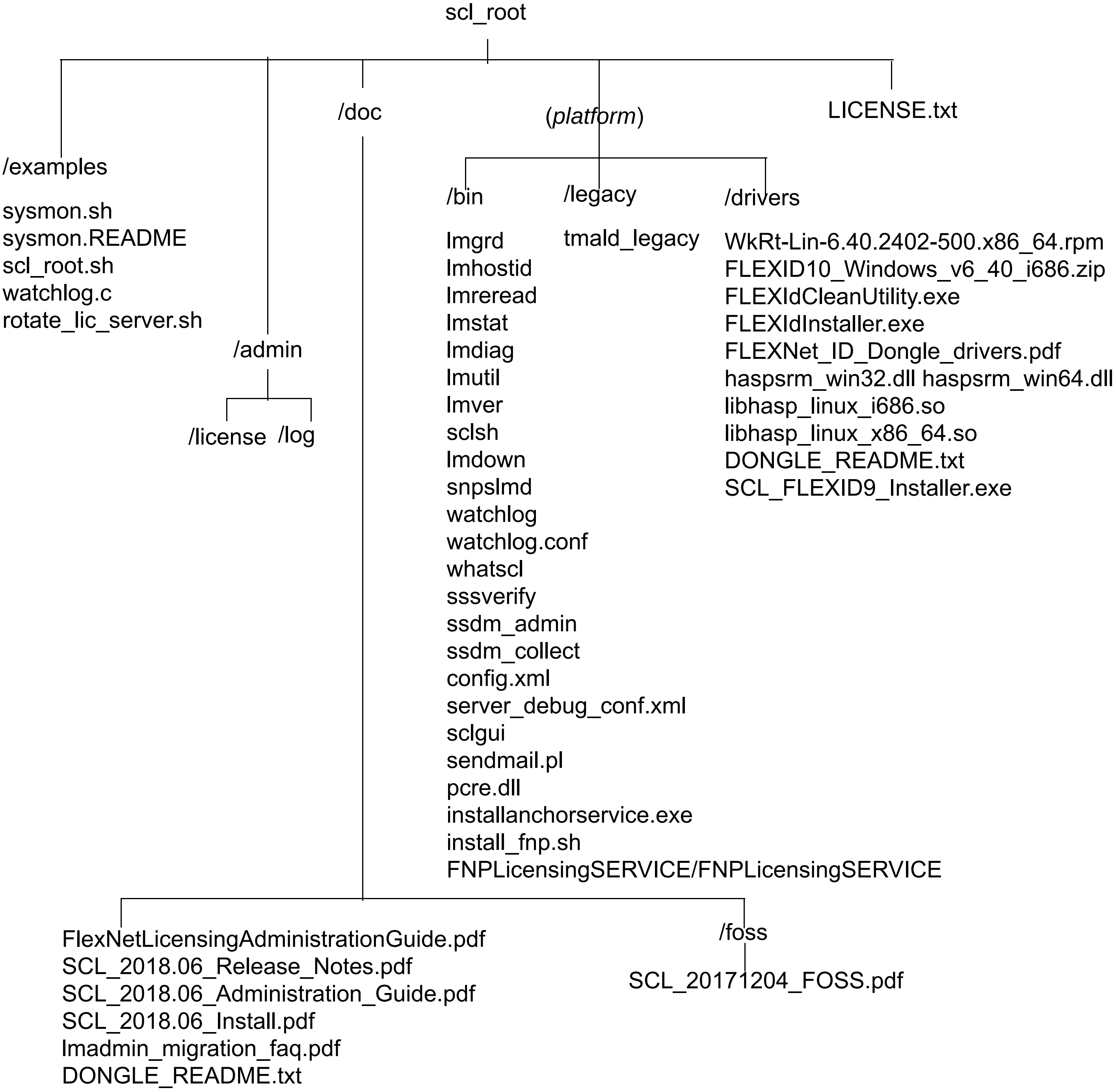


Table 5-1 describes the directories created by the installation program and the installed files.

Table 5-1 Installation Directory Structure and File Location and Description

Root and top-level directories	Subdirectories and files
scl_root	This is the root directory for the SCL software created by the installation program. This directory contains LICENSE.txt file and the following subdirectories: examples, admin, doc, *examples*, and *platform*.
examples	This directory includes setup files and example scripts that you can modify for your site, including a Linux startup script (which must be modified with site-specific information).
admin	The admin directory contains the following subdirectories: license and logs. • /license – This directory is provided as a possible location for the license key file. You can store the license key file where it is most convenient for your site. There is no default or required location. • /logs – The installation process creates this empty directory to which you can later install log files resulting from the `snpslmd` daemon processing.
doc	This directory contains the SCL documentation. It includes the following files: • FlexNetLicensingAdministrationGuide.pdf – This document contains information required by FLEXNet Licensing to manage licenses for a FLEX enabled application. • SCL_2018.06_Release_Notes.pdf – This document is the release notes that give brief installation and setup information as well as the latest updates about SCL. • SCL_2018.06_Administration_Guide.pdf – This document is the *Synopsys Common Licensing Administration Guide*. • SCL_2018.06_Install.pdf – This document is the *Synopsys Common Licensing Installation Notes*. • FLEXnet_ID_Dongle_Drivers.pdf – This document contains information about FLEXNet Dongle Drivers. • DONGLE_README.txt – This file contains additional information about the latest dongle driver installation instructions for Windows and Linux. • SCL_20171204_FOSS.pdf – This document contains licensing information about free and open-source software (FOSS) that is included with the Synopsys SCL product.

Table 5-1 Installation Directory Structure and File Location and Description (Continued)

Root and top-level directories	Subdirectories and files
drivers	This directory contains the various dongle drivers required to use a dongle as a license server. • WkRt-Lin-6.40.2402-500.x86_64.rpm – This file is a FLEXID=9 USB dongle driver installer for Red Hat Enterprise and SUSE Linux platforms (RHEL/SUSE). • FLEXID10_Windows_v6_40_i686.zip – This file is a FLEXID=10 WibuKey dongle driver for Windows platforms. • FLEXIdCleanUtility.exe – This file is the dongle driver cleanup and uninstall tool. • FLEXIdInstaller.exe – This file is the previous version of the FLEXID dongle driver installer. • FLEXNet_ID_Dongle_drivers.pdf – This document contains information about FlexNet Dongle Drivers. • haspsrm_win32.dll – This is a third-party dongle driver (DLL) file. It should be copied to the C:\Windows\SysWOW64 directory on a 64-bit Windows OS machine or to the C:\Windows\System32 directory on a 32-bit Windows OS machine. • haspsrm_win64.dll – This is a third-party dongle driver (DLL) file. It should be copied to the C:\Windows\System32 directory on a 64-bit Windows OS machine. • libhasp_linux_i686.so – This file is a third-party dongle shared library that needs to be copied to the /usr/lib for 32-bit Linux OS platforms. • libhasp_linux_x86_64.so – This file is a third-party dongle shared library that needs to be copied to the /usr/lib for 64-bit Linux OS platforms. • DONGLE_README.txt – This file contains additional information about the latest dongle driver installation instructions for Windows and Linux. • SCL_FLEXID9_Installer.exe – This file installs the FLEXID9 dongle drivers and copies the haspsrm_win*.dll to the correct system location depending on the Windows OS. See the *FlexNet Publisher Driver Installation Guide for FlexNet ID Dongles* for more information.

Table 5-1 Installation Directory Structure and File Location and Description (Continued)

Root and top-level directories	Subdirectories and files
<platform>	This directory contains the following subdirectories /bin – For each supported platform architecture (after which the directory is named), the bin subdirectory contains the same set of executable files. This directory contains the following SCL executable files, including the SCL daemons and utilities: • `lmgrd` – This executable file is the standard FlexNet license server daemon. • `snpslmd` – This executable file is the SCL vendor daemon. • `lmutil` • `lmver` • `lmstat` • `lmreread` • `lmhostid` • `lmdown` • `sclsh` – This executable file is the SCL shell, which you can use to test the licensing daemon. • `whatscl` – This executable file returns the version of SCL that you are running. • `sssverify` – This executable file verifies the integrity of your license file. • server_debug_conf.xml – This is the default configuration file used by SCL for custom debug logging. • `ssdm_admin` – This executable gathers the license server related performance data. For details on ssdm, see Synopsys Statistical Debug Manager (SSDM). • `ssdm_collect` – This executable is used by the ssdm_admin executable to collect information about the license server. • config.xml – This default configuration file is used by the ssdm executables.

Table 5-1 Installation Directory Structure and File Location and Description (Continued)

Root and top-level directories	Subdirectories and files
platform	• `watchlog` – This executable is used to monitor custom messages from the server log. It is supported on all Linux and Windows platforms. • watchlog.conf – This is the default configuration file used by the watchlog binary. • sclgui – This GUI tool performs the following operations: - License file Diff: Generates a diff operation between two license files based on specified parameters - Sort Keys: Sorts the license keys based on specified fields - Product to Keys: Generates the Product to Feature mapping report - Verify File: performs license file validation for authenticity and correct use of the license file • `sendmail.pl` (Windows only) – This Perl script sends e-mail alerts using the SMTP server. • pcre.dll (Windows only) – This file is used by the watchlog executable during runtime. • installanchorservice.exe (Windows only) – This executable file starts the FlexNet Licensing service on the SCL license server host machine to detect the host ID of the virtual machine. • install_fnp.sh (Linux only) – This script file starts the FlexNet Licensing service to detect host ID of the virtual machine. • FNPLicensingSERVICE/FNPLicensingSERVICE (Linux only) – This binary file is used by the install_fnp.sh script to start the FlexNet Licensing service.

You should use the Synopsys Installer to install the SCL software. For SCL configuration and start-up instructions, see Chapter 6, “Starting the SCL Server on Linux and Windows Systems.”

To download SCL via Electronic Software Transfer, download the scl_v2018.06_common.spf file and the platform specific tar file that you require. For example, for the x86_64 platform, you should download the scl_v2018.06_linux64.spf or the scl_v2018.06_linux.spf file.

The Synopsys Installer defaults to the following root directory:

```
/usr/synopsys/SCL/2018.06
```

Subdirectories are under this directory, such as:

```
/usr/synopsys/SCL/2018.06/linux64/bin/snpslmd
```

For the Windows platform, the default install directory is `C:\Synopsys\SCL\2018.06`

After installation, the directory looks like this:

```
C:\Synopsys\SCL\2018.06\win32\bin\snpslmd.exe
```

Synopsys also provides various dongle drivers for Windows and Linux platforms, which are required to use a dongle as a license server on Windows.

Drivers for the Windows x86_64 platform are included under this directory:

```
C:\Synopsys\SCL\2018.06\win32\drivers
```

Drivers for the Linux x86_64 platform are included under this directory:

```
/usr/synopsys/SCL/2018.06/linux64/drivers
```

Verifying the License Key File

This section describes how to determine if your license key file includes the correct host ID and how to verify that the license key file is valid.

The following sections are included:

- Verifying the License Key File Host ID
- Checking the License Key File for Errors
- Avoiding License Key File Errors
- Placing the License Key File on Your System

Verifying the License Key File Host ID

Use the `lmhostid` utility included with SCL to verify the license server host ID.

Linux

In the following example for the Linux x86_64 platform, the host ID of the machine is 001f296516d8:

```
/usr/synopsys/scl/2018.06/linux64/bin>  lmhostid

lmhostid - Copyright (c) 1989-2017 Flexera Software LLC. All Rights
Reserved.
The FLEXnet host ID of this machine is "001f296516d8"
```

Windows

In the following example, the host ID of the machine is 000f1f2fae0b:

```
C:\Synopsys\SCL_2018.06\win32\bin>  lmhostid
```

```
lmhostid - Copyright (c) 1989-2017 Flexera Software LLC. All Rights
Reserved.
The FLEXnet host ID of this machine is "000f1f2fae0b"
```

If the host ID number of your license server does not match the host ID specified on the SERVER line, locate the correct server or if there is no such server, contact your local sales representative.

Checking the License Key File for Errors

You must use the sssverify utility to check the license key file for errors before you install it. Errors in the license key file can result in license acquisition failures or denials. For detailed information about sssverify, see Checking the Integrity of the License File.

Before running sssverify, check the license key file for errors. At a minimum, check the following lines:

SERVER line: Make sure the SERVER line specifies the correct host name for the host ID. The host name is the first parameter; the host ID is the second parameter. Errors on the SERVER line can cause startup failures.

Be sure to change the specified TCP port (27020) if your site is using another TCP port for SCL (lmgrd). For details on the SERVER line, see The SERVER Line.

VENDOR line: Make sure the VENDOR line correctly specifies the SCL snpslmd daemon, followed by the full path to the snpslmd daemon. Errors on the VENDOR line can cause startup failures.

If you modified this line to include a path to the options file for the DAEMON line, ensure that the daemon is correct. For details about VENDOR line requirements, see The VENDOR Line.

Avoiding License Key File Errors

The following rules are mandatory. To avoid errors in the license key file, it is necessary that you follow these guidelines:

- Always run the sssverify utility after modifying the license file.
- Always treat the license file as a unit:
  - Never modify your license file by adding or removing licenses from another file, including keys from old Synopsys licenses or another vendor. (The only exception to this rule is temporary keys, which can be appended to your production key file if they are not invalidated by production keys with a later ISSUED date.)

  - ❍ Other than modifiable fields (the VENDOR path and the SERVER name or port number), do not add any information to the license key file.
  - ❍ Do not combine expired keys with the new key file. Combining older production keys with newer production keys invalidates the license key file and result in checkout errors or denials.

- Always use the latest versions of FlexNet daemons and utilities packaged with SCL. Never use `lmgrd` from another vendor or if from an earlier release of a Synopsys tool. Using older versions from other vendors or old Synopsys installs can cause licensing failures.
- SCL and all components (the license key file, options file, and debug log file) must be locally mounted.

Placing the License Key File on Your System

You can install the license key file in any locally mounted directory. However, when you install the SCL software, the installation program creates a directory called *`scl_root`*`/admin/license`. This is the recommended location for the license key file.

The license key file identifies the server to be used by the application. For Synopsys tools to find the license server, you must specify its location by using one of the environment variables. See Describing the License Environment Variables.

Required License Key File Modifications

The license key file is an ASCII text file that contains licensing data required for SCL to serve licenses. You can use the license key file after modifying the host name value on the SERVER line and the path information on the VENDOR line.

For a detailed explanation of the license key file, see Explaining the License Key File or the *License Administration Guide*.

Warning:

> Do not change any data items in the license key file other than those identified in this section as modifiable. Host ID information, as well as INCREMENT and PACKAGE lines, are encrypted. Changing these items invalidates the license keys.

The following topics are included in this section:

- Modifying the SERVER Line
- Modifying the VENDOR Line

Modifying the SERVER Line

You can modify the following data items on the SERVER line; otherwise, default values are assumed:

- The *host name*

  You must change the default host name `hostname1` supplied as a placeholder by Synopsys, replacing it with the actual host name for your server node.

  Note:
  For redundant servers, the license key file has three SERVER lines specifying hostname1, hostname2, and hostname3.

  The host name argument is a string returned by the Linux `hostname` or `uname -n` command, or an IP address in the ###.###.###.### format. When you edit this field, use the IP address if the server node has multiple network interfaces (and therefore, multiple host names), or for sites where the Network Information Service (NIS) or domain name structure (DNS) has trouble resolving a host name.

- The *port number*

  The lmgrd *port number* you receive from Synopsys specifies a default TCP port number of 27020. To use a different port number other than the default, specify a default lmgrd TCP port number within the range of 1024 through 64000 inclusive. Do not use ports 27000-27009 for lmgrd as these ports have higher latency.

  After you install and run the licensing software, if you modify the server host name or the TCP port number of the server, you must shut down the licensing system by using `lmdown` and restart the `lmgrd` daemon to enact these changes. (The `lmreread` utility does not cause these specific changes to take effect.)

Modifying the VENDOR Line

Synopsys tools require an active license server daemon at all times. SCL includes the standard `lmgrd` FlexNet license server daemon and the `snpslmd` Synopsys vendor daemon.

The `snpslmd` daemon supports all Synopsys vendor daemons. Therefore, only one vendor daemon is needed for all Synopsys tools.

A license server machine can have only one active `snpslmd` daemon at a time. No other Synopsys vendor daemons can run on the server with `snpslmd`.

For distributed license networks, you must start the `lmgrd` daemon on each license server. For instructions on starting the license server daemon, see Starting the SCL License Server Daemon.

The VENDOR line, specifies the `snpslmd` Synopsys daemon name and path and a daemon options file. The VENDOR line also allows you to specify a port number to be used if you make connections through an internet firewall. Many internet firewalls require that port numbers to the firewall be specified.

The `lmgrd` license server daemon uses the VENDOR line to locate and start the Synopsys `snpslmd` daemon. The Synopsys `snpslmd` daemon refers to this line for path information to its options file. (You can provide an options file to specify various operating parameters of the license server such as access control.)

Syntax

```
VENDOR daemon_name [daemon_path] \
         [options= options_path] [port= port_number]
```

Here is the VENDOR line as it exists unmodified in the license key file:

```
VENDOR snpslmd /path/to/snpslmd
```

Important:

You must specify the full path for the `snpslmd` vendor daemon on the VENDOR line.

Here is the same VENDOR line modified to give the path to the location where the `snpslmd` daemon is stored, as well as an options file location with a fully qualified path:

```
VENDOR snpslmd scl_root/linux64/bin/snpslmd \
options=scl_root/linux64/admin/snpslmd.opt
```

If you provide an options file and choose to accept the default location rather than explicitly specifying the file’s location, you must name the file `snpslmd.opt` and store it in the same directory as the license key file. For more details, see Locating and Using the Options File.

If you specify an options file, it is recommended that you use the `options=` keyword in conjunction with the options file path.

You can check the debug log file to verify that the license key file and options file were found and properly read. (You must have write access to the directory containing the debug.log file.)

For details about the debug log file, see Using the Debug Log File to Troubleshoot License Issues.

Locating and Using the Options File

Use of an options file enables a finer degree of access control over licensed features. You can use the file to allow or restrict access to licensed features for certain users or machines, based on user names, host names, display names, or IP addresses. You can also reserve licensed features using these same categories.

The `snpslmd` daemon looks at the `options=path` value on the VENDOR line in the license key file for a specified path. If you do not specify the path, the file must be named snpslmd.opt and must reside in the same directory as the license key file.

You should not invoke the `lmreread` command on a license server to read the changes made to the options file. Instead, restart the license server if you want the changes from the options file to be properly read by the license server.

All option file keywords must be specified in uppercase letters, for example, RESERVE, INCLUDE, EXCLUDE, and so on. Keywords specified in lowercase letters are not recognized.

If you have PACKAGE lines in the license key file, do not specify any license checkout policies such as `RESERVE`, `INCLUDE`, or `EXCLUDE`, in the options file for any package (or its components). Packages already have a license checkout policy, and this might conflict with the policy specified in the options file.

If set, the minimum value of the TIMEOUT or TIMEOUTALL options file keyword is 3600 (60 minutes).

6

Starting the SCL Server on Linux and Windows Systems

This chapter provides steps for starting the SCL server and information about the license environment variables.

The chapter includes the following sections:

- Starting the SCL License Server Daemon
- Specifying a Debug Log File
- Describing the License Environment Variables

Starting the SCL License Server Daemon

Synopsys tools require an active license server daemon at all times. SCL includes two daemons, the standard FlexNet lmgrd license server daemon and the snpslmd Synopsys vendor daemon.

The snpslmd daemon supports all Synopsys vendor daemons incorporated into SCL. Therefore, only one vendor daemon is needed for all Synopsys tools. (For a list of vendor daemons incorporated into SCL, see Synopsys Legacy Vendor Daemons.

A license server machine can have only one active snpslmd daemon at a time. No other Synopsys vendor daemons (including legacy daemons) can run on the server with snpslmd.

Starting the License Server on One or More Servers

This section describes how to start the SCL server from the command line and discusses the debug log file.

Using the lmgrd Command to Start SCL

You start SCL on a server by using the lmgrd command. This command automatically starts SCL by reading the license key file and starting the snpslmd vendor daemon.

For three-server redundancy, you must start the lmgrd daemon on each of the three servers listed in the SERVER lines of your license key file.

To start SCL 2018.06, follow these steps:

1. Modify the license key file. See Required License Key File Modifications.
2. Stop snpslmd and any legacy vendor daemons by using the lmdown utility included with SCL.

   ```
   % lmdown -c /path/to/synopsys.lic
   ```

   For Linux platforms, lmdown is installed in the *scl_root*/*platform*/bin directory. For Windows systems, it is installed in the *scl_root*\win32\bin directory. (For optional switches, see the *License Administration Guide*.)

   For the Windows OS, you can also use lmtools to stop the server.
3. Start the SCL 2018.06 license server.

   Note:
   For Windows users, it is recommended that you use lmtools to start SCL. For detailed instructions, see Starting SCL Automatically With LMTOOLS.

The following example (which starts `lmgrd` and specifies a debug log file) assumes the license key file is stored in the `scl_root/admin/license` directory:

```
% scl_root/platform/bin/lmgrd  \
-c scl_root/admin/license/snpslmd.lic \
-l scl_root/admin/logs/debug.log -reuseaddr
```

Replace *platform* with the keyword for your operating system. For a list of SCL keywords, see Table 1-1, Table 1-2, Table 1-3, and Table 1-4.

Note:
The `lmgrd` daemon locates and starts the Synopsys `snpslmd` daemon based on path information specified on the VENDOR line of the license key file.

4. Check the debug log file to make sure that the server has been started correctly.

   The debug log file contains output from the `lmgrd` and `snpslmd` daemons. For details and an example log file report, see Specifying a Debug Log File.

5. (Optional) Use `lmstat` to verify that users are able to check out licenses and to ensure that the license daemon started properly. Output from `lmstat` should show the FlexNet version and the name of the host used as the license server, followed by the word UP.

   The following lines indicate that the `snpslmd` daemon is active.

```
host1: license server UP (MASTER)v11.14.1
Vendor daemon status (on host1)
snpslmd: UP v11.14.1
```

Note:
The lmstat utility displays only the major FlexNet version, for example, 11.14.1. The minor version (11.14.1.3) is not displayed.

Specifying a Debug Log File

When you start the `lmgrd` daemon, you can specify a debug log file to which diagnostic information is directed. (The debug log file receives the output from the `lmgrd` daemon and the `snpslmd` daemon.) For example,

```
lmgrd –c synopsys.lic –l synopsys.log
```

You can specify any name for the output log file, for this example, the log file is, license.log. Note that you must have write access to the directory containing the debug log file.

Example 6-1 shows an example of debug log file output.

Example 6-1 Debug Log File Output

```
 0:06:03 (lmgrd) -----------------------------------------------
 0:06:03 (lmgrd)   Please Note:
 0:06:03 (lmgrd)
 0:06:03 (lmgrd)   This log is intended for debug purposes only.
 0:06:03 (lmgrd)   In order to capture accurate license
 0:06:03 (lmgrd)   usage data into an organized repository,
 0:06:03 (lmgrd)   please enable report logging. Use Flexera Software LLC's
 0:06:03 (lmgrd)   software license administration  solution,
 0:06:03 (lmgrd)   FlexNet Manager, to  readily gain visibility
 0:06:03 (lmgrd)   into license usage data and to create
 0:06:03 (lmgrd)   insightful reports on critical information like
 0:06:03 (lmgrd)   license availability and usage. FlexNet Manager
 0:06:03 (lmgrd)   can be fully automated to run these reports on
 0:06:03 (lmgrd)   schedule and can be used to track license
 0:06:03 (lmgrd)   servers and usage across a heterogeneous
 0:06:03 (lmgrd)   network of servers including Windows NT, Linux
 0:06:03 (lmgrd)   and UNIX.
 0:06:03 (lmgrd)
 0:06:03 (lmgrd) -----------------------------------------------
 0:06:03 (lmgrd)
 0:06:03 (lmgrd)
 0:06:03 (lmgrd) Server's System Date and Time: Wed Apr 25 2018 00:06:03 PDT
 0:06:03 (lmgrd) SLOG: Summary LOG statistics is enabled.
 0:06:03 (lmgrd) FlexNet Licensing (v11.14.1.3 build 212549 x64_lsb) started on
hostname1 (linux) (4/25/2018)
 0:06:03 (lmgrd) Copyright (c) 1988-2017 Flexera Software LLC. All Rights Reserved.
 0:06:03 (lmgrd) World Wide Web:  http://www.flexerasoftware.com
 0:06:03 (lmgrd) License file(s): test.lic
 0:06:03 (lmgrd) lmgrd tcp-port 27020
 0:06:03 (lmgrd) (@lmgrd-SLOG@) ===============================================
 0:06:03 (lmgrd) (@lmgrd-SLOG@) === LMGRD ===
 0:06:03 (lmgrd) (@lmgrd-SLOG@) Start-Date: Wed Apr 25 2018 00:06:03 PDT
 0:06:03 (lmgrd) (@lmgrd-SLOG@) PID: 881
 0:06:03 (lmgrd) (@lmgrd-SLOG@) LMGRD Version: v11.14.1.3 build 212549 x64_lsb ( build
212549 (ipv6))
 0:06:03 (lmgrd) (@lmgrd-SLOG@)
 0:06:03 (lmgrd) (@lmgrd-SLOG@) === Network Info ===
 0:06:03 (lmgrd) (@lmgrd-SLOG@) Listening port: 27020
 0:06:03 (lmgrd) (@lmgrd-SLOG@)
 0:06:03 (lmgrd) (@lmgrd-SLOG@) === Startup Info ===
 0:06:03 (lmgrd) (@lmgrd-SLOG@) Server Configuration: Single Server
 0:06:03 (lmgrd) (@lmgrd-SLOG@) Command-line options used at LS startup: -c test.lic
-l ser.log -reuseaddr
 0:06:03 (lmgrd) (@lmgrd-SLOG@) License file(s) used:  test.lic
 0:06:03 (lmgrd) (@lmgrd-SLOG@) ===============================================
 0:06:03 (lmgrd) Starting vendor daemons ...
 0:06:03 (lmgrd) Started snpslmd (internet tcp_port 36277 pid 883)
 0:06:03 (snpslmd) FlexNet Licensing version v11.14.1.3 build 212549 x64_lsb
04/25/2018 00:06:03 (snpslmd) INFO: Pooling on VENDOR_STRING is Disabled (to enable,
set SCL_POOL_VS=1)
04/25/2018 00:06:03 (snpslmd) Synopsys Corporate Licensing (SCL) Release: version
SCL_2018.06
04/25/2018 00:06:03 (snpslmd) [SCL] File descriptor limit (soft, hard) for snpslmd:
1024, 1024
04/25/2018 00:06:03 (snpslmd) INFO: [SCL] TCP - tcp_keepalive_time - 7200 |
tcp_keepalive_intvl - 75 | tcp_keepalive_probes - 9 | tcp_fin_timeout - 60
 0:06:03 (snpslmd) SLOG: Summary LOG statistics is enabled.
 0:06:03 (snpslmd) SLOG: FNPLS-INTERNAL-CKPT1
 0:06:03 (snpslmd) SLOG: VM Status: 0
 0:06:03 (snpslmd) Server started on hostname1 for:DC-Expert
 0:06:03 (snpslmd) DC-Ultra-Features DC-Ultra-OptDesign-Compiler Designware
 0:06:03 (snpslmd) HDL-CompilerISE-TCAD_feat1all_feat
 0:06:03 (snpslmd) SSS
 0:06:03 (snpslmd)
 0:06:03 (snpslmd) Licenses are case sensitive for TE_CATS
 0:06:03 (snpslmd)
 0:06:03 (snpslmd) EXTERNAL FILTERS are OFF
 0:06:03 (lmgrd) snpslmd using TCP-port 36277
 0:06:03 (snpslmd) Serving features for the following vendor names:
 snpslmd  ACAD  adalmd  anagram  arcd  atrenta  avantd  CADABRA  chrysalisd  cowared
EPIC  eved  extremed  hscd  innologd  ISE-TCADd  knights  magma  metasoftd  mwflexd
nassd  nsysnvs  numeritchd  pdld  perflmd  riod  saber_dmn  sandwork  sclmgrd  sigmacd
SIINT  simple  snpsOEM1  snslmgrd  ssilmd  synforad  synopsysd  synplctyd  targetld
TAVEREN  TE_CATS  tmald  vastlmd  vcsd  wlld
04/25/2018 00:06:03 (snpslmd)
------------------------------------------------------------------
04/25/2018 00:06:03 (snpslmd) Checking the integrity of the license file...
04/25/2018 00:06:03 (snpslmd) Valid SSS feature found.
04/25/2018 00:06:03 (snpslmd) The feature is needed to enable the other keys in your
license file.
04/25/2018 00:06:03 (snpslmd) Licensed to Synopsys R & D
04/25/2018 00:06:03 (snpslmd) Siteid: xxxxx, Server Hostid: xxxxx, Issued on: 4/25/
2018
04/25/2018 00:06:03 (snpslmd)
------------------------------------------------------------------
 0:06:03 (snpslmd) SLOG: Statistics Log Frequency is 240 minute(s).
 0:06:03 (snpslmd) SLOG: TS update poll interval is 0. TS update is detected by
midnight reread only.
 0:06:03 (snpslmd) SLOG: Activation borrow reclaim percentage is 0.
 0:06:03 (snpslmd) (@snpslmd-SLOG@) ===============================================
 0:06:03 (snpslmd) (@snpslmd-SLOG@) === Vendor Daemon ===
 0:06:03 (snpslmd) (@snpslmd-SLOG@) Vendor daemon: snpslmd
 0:06:03 (snpslmd) (@snpslmd-SLOG@) Start-Date: Wed Apr 25 2018 00:06:03 PDT
 0:06:03 (snpslmd) (@snpslmd-SLOG@) PID: 883
 0:06:03 (snpslmd) (@snpslmd-SLOG@) VD Version: v11.14.1.3 build 212549 x64_lsb (
build 212549 (ipv6))
 0:06:03 (snpslmd) (@snpslmd-SLOG@)
 0:06:03 (snpslmd) (@snpslmd-SLOG@) === Startup/Restart Info ===
 0:06:03 (snpslmd) (@snpslmd-SLOG@) Options file used: None
 0:06:03 (snpslmd) (@snpslmd-SLOG@) Is vendor daemon a CVD: Yes
 0:06:03 (snpslmd) (@snpslmd-SLOG@) Is TS accessed: No
 0:06:03 (snpslmd) (@snpslmd-SLOG@) TS accessed for feature load: -NA-
 0:06:03 (snpslmd) (@snpslmd-SLOG@) Number of VD restarts since LS startup: 0
 0:06:03 (snpslmd) (@snpslmd-SLOG@)
 0:06:03 (snpslmd) (@snpslmd-SLOG@) === Network Info ===
 0:06:03 (snpslmd) (@snpslmd-SLOG@) Listening port: 36277
 0:06:03 (snpslmd) (@snpslmd-SLOG@) Daemon select timeout (in seconds): 1
 0:06:03 (snpslmd) (@snpslmd-SLOG@)
 0:06:03 (snpslmd) (@snpslmd-SLOG@) === Host Info ===
 0:06:03 (snpslmd) (@snpslmd-SLOG@) Host used in license file: hostname1
 0:06:03 (snpslmd) (@snpslmd-SLOG@) Running on Hypervisor: Not determined - treat as
Physical
 0:06:03 (snpslmd) (@snpslmd-SLOG@) ===============================================
```

Describing the License Environment Variables

This section describes the environment variables that you set on every client to identify the location of the SCL license server to be used. It also describes how to set the environment variables by using *port@host* syntax.

SCL supports use of the FlexNet vendor-specific license environment variable implemented as `SNPSLMD_LICENSE_FILE` and the standard FlexNet `LM_LICENSE_FILE` variable.

Note:
For additional information about using environment variables on Windows platforms, see Starting SCL on Windows Platforms With LMTOOLS.

About Environment Variables

To locate the SCL license server to be used, each client must have one of its license environment variables set to point to the software. The license environment variables refer to one or more license servers (or, in the outdated mode, license key files) to be used.

SCL uses the following two license environment variables, common to all Synopsys tools:

- `SNPSLMD_LICENSE_FILE`

  This is the Synopsys specific (vendor) license environment variable. It is recommended that you use the `SNPSLMD_LICENSE_FILE` environment variable because the latest Synopsys tools read this environment variable first. License requests can be handled faster if the Synopsys tools do not have to contact servers from other vendors.

  You can assign multiple values to the `SNPSLMD_LICENSE_FILE` variable, but all of these values should refer to SCL license servers, not to other vendor products. If you set this variable to multiple values, the license key files on the servers listed are searched from left to right.

- `LM_LICENSE_FILE`

  The `LM_LICENSE_FILE` license variable, which is the FlexNet default, can be used to refer to multiple license key files, including those provided by vendors other than Synopsys. If multiple file locations are assigned to this variable, the Synopsys tools search those files from left to right to find the requested license.

  Important:
  To speed tool startup, some Synopsys tools do not look at `LM_LICENSE_FILE` if `SNPSLMD_LICENSE_FILE` is set.

Specifying Values for License Software Environment Variables

Although there are several methods you can use to identify the license server to be used, using the *port@host* method for all SCL clients is recommended.

Note:

Specifying an explicit path or directory (the directory containing the license key file or files with the .lic extension) is not supported. Use the *port@host* method.

Table 6-1 compares the various methods of specifying license key environment variable values.

Table 6-1 Overview of Methods of Specifying Environment Variable Value

Notation	Advantages	Disadvantages
port@host	Recommended method. Uses the port number and host name specified on the SERVER line of the license key file. Makes available to the client any updates to the license key file.	A TCP port number for the license server must be explicitly specified in the license key file. You can use the Synopsys default TCP port number of 27020 or choose a number from within the range of 1024 - 64000 inclusive.
explicit path or directory	Not supported. Use the port@host method.	The file must be mounted, which can cause latency problems and checkout failures.

The port@host Notation

The recommended method of specifying the license server is to use the port@host notation. In this case, you use the port and host values specified on the SERVER line in the license key file.

The following example specifies a server with the host name of `host1` and a TCP port of 27020.

```
% setenv SNPSLMD_LICENSE_FILE 27020@host1
```

Using Environment Variables to Configure Clients for Multiple-Server Access

To configure a client for access to multiple distributed license servers, you set the environment variable to a license file list. All Synopsys tools have the default license file list behavior.

You can configure a client to refer to multiple servers for its license requests. To do so, you assign a list of servers to the license environment variable on the client server. The servers are queried in the order you specify them. If the requested license is not available on the first server, the next one is tried, and so on.

After a client is served a license from a particular server, it must obtain any additional licenses it requests from that server for the duration of the process unless the client can use license server pooling.

To configure a client to refer to its license servers, set the `SNPSLMD_LICENSE_FILE` or `LM_LICENSE_FILE` environment variable to refer to the list of servers to be queried for the license.

Example 6-2 uses the port@host specification to identify the single server to be queried for licenses for Client1:

Example 6-2 Querying Licenses for Single Server

```
% setenv SNPSLMD_LICENSE_FILE 27020@Server1
```

If the client has more than one license server, use a colon (:) as the delimiter to set the `SNPSLMD_LICENSE_FILE` or `LM_LICENSE_FILE` environment variable to refer to the list of servers to be queried for the license. Example 6-3 uses the port@host specifications to identify the three servers to be queried for licenses for Client2:

Example 6-3 Querying Licenses for Multiple Servers

```
% setenv SNPSLMD_LICENSE_FILE 27020@Server1:26585@Server2:28000@Server3
```

Note:

For Windows operating systems, use a semicolon (;) rather than a colon (:) as the delimiter.

Using Environment Variables to Configure Clients for Three-Server Redundancy

If your license network uses three-server redundancy, (described in Using Environment Variables to Configure Clients for Multiple-Server Access), server redundancy takes effect automatically. If the master server fails, another master server is chosen without the need to modify the configuration.

To use a redundant server configuration, specify all the redundant server machines using a comma (,) as the delimiter. For example,

```
setenv SNPSMD_LICENSE_FILE
27020@Server1,27020@Server2,27020@Server3:26585@remote_server1:28000@remo
te_server2
```

Warning:

> Use of redundant servers is not recommended. This method is not very robust and can cause the server system to fail if communication is lost between the servers.

Using the lmdown Command to Stop SCL

To stop SCL on a server, you can use the `lmdown` command. This command shuts down the `lmgrd` license server daemon and the `snpslmd` vendor daemon. (For three-server redundancy, you must stop the `lmgrd` license server vendor daemon on each of the three servers listed in the SERVER lines of your license key file).

To stop SCL version 2018.06, stop the `snpslmd` vendor daemon and any legacy vendor daemons by running the `lmdown` utility that is included with SCL using the following format:

```
% lmdown -c /path/to/synopsys.lic
```

For Linux platforms, the `lmdown` utility is installed in the scl_root/platform/bin directory. For Windows systems, it is installed in the scl_root\win32\bin directory. Alternatively, for the Windows operating system, you can use `LMTOOLS` to stop the server.

When you have successfully stopped SCL, the debug log file indicates that the daemon is not active.

Note:

> Using the `lmreread` utility is not supported. The `lmreread` utility can put the server in an unpredictable state, especially when there are options file restrictions on large license files or on redundant servers.

7

Explaining the License Key File

This chapter gives an overview of the Synopsys Common Licensing (SCL) license key file. It includes an example file in the format in which you receive it from Synopsys and provides a description of its contents.

This chapter includes the following sections:

About the License Key File

Licensing data required for SCL to grant (or deny) licenses is stored in a text file called the license key file. Synopsys generates the license key file based on the licenses purchased for tools served by a host or all hosts at the specified site.

To obtain the latest key file for your site, log on to SmartKeys at

https://solvnet.synopsys.com/SmartKeys

For details on retrieving the license key file through SmartKeys see Logging In to SmartKeys.

SCL allows you to use your license key file with minimal modification. You need only modify the SERVER line (to replace `hostname1` with the name of the license server) and the VENDOR line (to replace `/path/to/snpslmd` with the correct path to the `snpslmd` vendor daemon).

The license key file is designed to use default values for all user-modifiable data items. You can accept default values for the `lmgrd` TCP port (27020) and the `snpslmd` TCP port (determined randomly if not specified).

In addition, the license key file for SCL

- Uses the INCREMENT line rather than the FEATURE line to specify information
- Is case-sensitive
- Reports error messages that are clear and easy to understand

The License Key File Header

Example 7-3 shows a license key file which you would receive from Synopsys. The example file includes a monthly technology subscription license, as well as term licenses. Technology subscription licenses are regular licenses whose duration—defined by an explicit expiration date—is variable, but commonly extends for a year. Monthly technology subscription licenses have a usage limit of 30 days.

This section includes the following topics:

- License Key File Comment Lines
- The SERVER Line
- The VENDOR Line
- The USE_SERVER Line

License Key File Comment Lines

The license key file content is summarized in comment lines at the top of the file so you can see at a glance the kind and number of licenses issued. (The comment lines start with a pound sign (#) and can be removed.)

The following information is provided in the license file comments:

Product

The material and serial numbers identifying the product, followed by the name of the product for which the licenses are issued. The first number is the material number and is followed by the serial number. Each product has its own material number. Also, a product for which a license was issued in May has a serial number different from the same product for which a license was issued in January.

Qty

The number of licenses issued for the particular product.

Sup_start

The beginning date of the duration of support purchased for the product.

Sup_end

The end date of the duration of support purchased for the product.

Lic_Type

The type of licenses purchased. Example 7-3 shows two of the possible types of licenses: time based and perpetual.

Lic_Mode

The mode for which the license was issued. In Example 7-3, all licenses are networked licenses.

The SERVER Line

The first uncommented line in a license key file is the SERVER line. This line identifies the host node to be used as the server.

The SERVER line from the license key file shown in Example 7-3 follows:

```
SERVER hostname2 001f296516d8 27020
```

The SERVER line specifies the following fields:

`SERVER` (keyword)

hostname (modifiable)

> This is the name of the host machine used as the server. The host name, `hostname1`, is a default (placeholder) name. You must modify the default host name (`hostname1`) by replacing it with the actual name of the license server host machine.

host_ID (not modifiable)

> This is the host ID of the machine to be used as the server. This is the value you communicated to Synopsys before receiving your license key file.
>
> If you receive a license key file generated for an incorrect host ID, contact your sales representative.
>
> Warning:
>
> > Do not change the host ID value because it is used in the license key encryption process. The host ID value in the license key file must match the host ID value of the physical machine as determined by the `lmhostid` command; otherwise the license key file is invalid.
>
> The host ID of the server specified in the example SERVER line is 001f296516d8. For more information about the host ID, see Chapter 4, “Downloading SCL Software and Obtaining License Key Files.”

port_number (modifiable)

> The TCP port number used by the `lmgrd` daemon.
>
> For the example SERVER line, the TCP port number for the `lmgrd` daemon running on that server uses the default 27020. Valid values are between 1024 and 64000 inclusive. For performance reasons, using the default lmgrd ports 27000 through 27009 is not recommended.

The VENDOR Line

SCL uses the VENDOR line to identify the location of the `snpslmd` vendor daemon. (In other licensing systems based on FlexNet, this line might be called a DAEMON line.)

The VENDOR line in the license key file shown in Example 7-3 follows:

```
VENDOR snpslmd /path/to/snpslmd 27020
```

The VENDOR line specifies the following information:

VENDOR *daemon_name daemon_path* {options_file_path} \
{port=*tcp_port*}

daemon_name (not modifiable)

This is the name of the Synopsys vendor daemon, `snpslmd`.

Do not modify this value.

daemon_path (modifiable)

This is the path to the Synopsys `snpslmd` daemon.

options_file_path ()

This is the fully qualified path to the options file used by the Synopsys `snpslmd` daemon. (The license key file VENDOR line in Example 7-3 does not specify an options file path.)

Note:

If you require an options file, it is recommended that you specify the path to the options file. For example,

```
/path/to/snpslmd.opt
```

or

```
options=/path/to/snpslmd.opt
```

If you do not specify the path, the file must be named snpslmd.opt and must reside in the same directory as the license key file.

tcp_port (optional)

This is the fixed TCP port number for the vendor daemon to use, (for example, port=27021). This port, if specified, must be different than the port specified on the SERVER line.

This value is required only if clients need to connect to the license server from outside the firewall. (Many internet firewalls require that internal port numbers accessed from outside the firewall be specified.)

In the license key file shown in Example 7-3, the VENDOR line does not specify a TCP port number for the `snpslmd` daemon. Therefore, the TCP port number to used by `snpslmd` is determined randomly.

The USE_SERVER Line

By default, every Synopsys license key file includes a `USE_SERVER` keyword after the VENDOR line. Do not remove this keyword. The USE_SERVER line prevents client applications from reading the license file and forces them to request their licenses by network calls to the server specified on the SERVER line.

The USE_SERVER line from the license key file shown in Example 7-3 follows:

```
USE_SERVER (keyword)
```

The License Key File INCREMENT Line

The INCREMENT line defines the licenses (features) that are included with a product.

The INCREMENT line from Example 7-3 follows:

```
INCREMENT PrimeTime snpslmd 2017.06 31-dec-2018 3 VENDOR_STRING=^1+S \
   SUPERSEDE ISSUED=08-jun-2015 ck=79 SN=RK:2595-0:868738:325472 \
   START=15-apr-2014 AUTH={ snpslmd=( LK=A3A249E430FDB5B4) }
```

The remainder of this section explains the fields comprising the INCREMENT line.

Important:
The following fields cannot be modified.

feature_name

The name of the feature to which the license pertains. In the example, PrimeTime is the feature.

vendor_daemon

The name of the vendor daemon. For SCL, the vendor daemon is always `snpslmd`.

version

The version number of the feature supported by this license. The example specifies a license for version 2017.06 of PrimeTime. The version field is typically in date format, where 2017 corresponds to the year of the tool release and 06 to the sixth month of the year.

exp_date

The date when the specified license expires. In Example 7-3, the PrimeTime license expires on January 31, 2018 (31-jan-2018).

number_of_licenses

The number of concurrent licenses for the feature. In the example, three licenses of PrimeTime can be concurrently checked out.

`VENDOR_STRING`

The `VENDOR_STRING` in the example INCREMENT line has a value of `^1+S`, indicating that the PrimeTime feature is on support. This string is present only in on-support keys and provides forward-compatibility for most Synopsys products. Synplicity products do not use the on-support functionality.

For details about on-support keys, see On-Support Keys.

`SUPERSEDE`

Specifies that all licenses issued before the date specified by the `ISSUED` field are superseded by this line and invalidated. For details about SUPERSEDE, see SUPERSEDE Keys.

`ISSUED`

The date the license was issued. The example license for PrimeTime was issued on June 8, 2015 (ISSUED=08-jun-2015).

`ck`

This is a checksum field [ck=nnn] but is no longer used.

`SN`

The `SN` (serial number) string contains various subfields.

The first subfield is set to one of these values:

- RK (revenue key) denotes a technology subscription license.
- TK (temporary key)

The serial number for the PrimeTime license key shown in the previous example is SN=RK:2595-0:868738:325472. In this example, the first subfield is set to RK.

The second subfield is the material number, which represents the product that this feature belongs to. In the example, the material number is 2595-0, which corresponds to the product PrimeTime.

The third subfield is a security field.

START

The date when the license can first be used. (This also corresponds to the start date of the license period). For the example INCREMENT line, the start date for use of the PrimeTime license is April 15, 2014 (START=15-apr-2014).

AUTH

This field contains a list of supported daemons as well as encryption information. In the example key, the AUTH field looks like this:

```
AUTH={ snpslmd=(LK=A3A249E430FDB5B4)}
```

This means that this key supports clients (tools) built only on the snpslmd daemon. The LK= fields contain encryption information.

On-Support Keys

On-support keys are licenses that contain the string VENDOR_STRING=^1+S. On-support licenses provide continuous support for product releases up to two years beyond the version specified in the version field. The advantage of on-support licenses is that you do not need to upgrade your license file for every new release.

Consider the following INCREMENT line for the Design Compiler feature shown in Example 7-1.

Example 7-1 On-Support License Example

```
INCREMENT Design-Compiler snpslmd 2018.06 31-dec-2018 9 \
   VENDOR_STRING=^1+S SUPERSEDE ISSUED=21-jun-2018 \
   ck=50 SN=RK:2661-0:668133:325472 START=15-apr-2012 \
   AUTH={ snpslmd=( LK=5D86A4B97689DA46C550)}
```

This on-support license is valid for Design Compiler versions through 2020.06, which is two years later than the version field (2018.06). Note that this key cannot be used after the expiration date of December 31, 2018.

If this license did not have the on-support string (VENDOR_STRING=^1+S), you could not use a later version of Design Compiler, such as 2019.03.

Note:

On-support revenue keys display the version in the output of the lmstat command different from the version number of the tool being used. In Example 7-1, lmstat lists the tool version as 2018.06, but in the list checkouts show 2018.06 tool users as using version 2016.06. This is done to enable the "on-support" two-year forward-compatibility functionality described in Example 7-1.

SUPERSEDE Keys

SUPERSEDE keys have a special SUPERSEDE field. SUPERSEDE keys void all licenses (with the same feature names) that have an issue date earlier than the SUPERSEDE key's ISSUED date.

Consider the SUPERSEDE keys license file in Example 7-2:

Example 7-2 SUPERSEDE Keys License File Example

```
INCREMENT PrimeTime snpslmd 2018.06 31-dec-2018 8 VENDOR_STRING=^1+S \
   SUPERSEDE ISSUED=21-jun-2018 ck=4 SN=RK:2595-0:868738:325472 \
   SN=RK:2595-0:868738:325472 \
   START=30-jan-2014 AUTH={ snpslmd=( LK=9D76D499C7EE3BAC14EF) }

INCREMENT PrimeTime snpslmd 2018.06 20-oct-2018 2 \
   ISSUED=20-jun-2018 ck=4 SN=TK:2595-0:0:211426:517493\
   START=20-jun-2018 AUTH={ snpslmd=( LK=9D76D499C7EE3BAC14EF) }
```

Example 7-2 includes two PrimeTime keys (INCREMENT lines), one revenue (technology subscription license) key (SN=RK) and one temporary key (SN=TK). These keys support a total of 10 seats (8 + 2).

The first PrimeTime key has a SUPERSEDE field and an issue date of June 21, 2018. This key voids any PrimeTime keys issued before that date. Because the second PrimeTime key has an earlier issue date (ISSUED=20-jun-2018), it is voided. Therefore, only 8 seats of PrimeTime are available.

All license keys for technology subscription licenses (production licenses) include the SUPERSEDE keyword. Typically, temporary licenses do not include SUPERSEDE.

Synopsys Example License File

Example 7-3 shows a typical license key file in the format delivered by Synopsys. The example file includes on-support technology subscription licenses, an on-support perpetual (20-year) license, and an on-support monthly technology subscription (peak) license. (This example does not include any temporary licenses because they are not delivered by SmartKeys.)

Example 7-3 Synopsys License Key File Example

```
# Synopsys License file created by CVDKIF 3.5.1
SERVER hostname 001f296516d8 27020
VENDOR snpslmd /path/to/snpslmd
USE_SERVER

INCREMENT SSS snpslmd 1.0 19-Oct-2031 1 B2135255BFE7 \
   VENDOR_STRING="eb89e 9fafb 8be2c 9030a 38ed3 0d014 db251 c4774 b626d f0a11 \
   01020 02365 21474 d96a0 16150 ed114 9260e e64ed c8a8f ae72c \
   6cec8 238de 70f04 fca07 1b94a 75dd3 cc295 35a35 04a95 ebcf2 \
    ....................................................................................................................................................\
    6ba03 7eac0 02d8b 81a" \
    NOTICE="Licensed to Synopsys, Inc. [DO NOT DELETE/MODIFY SSS OR ANY \
    OTHER KEYS IN THIS FILE]" \
    SN=RK:0:0:823872 ISSUER="Synopsys, Inc. [06/21/2013 12:44:49 100]" \
    START=20-Oct-2011

INCREMENT DesignWare snpslmd 2012.12 31-dec-2013 VENDOR_STRING=^1+S \
   SUPERSEDE ISSUED=21-jun-2010 ck=253 SN=PK:2925-0:847487:9396:475084\
   START=20-oct-2011 AUTH={ snpslmd=(LK=1D1675154904BAC87FBD) }

INCREMENT DesignWare snpslmd 2013.03 19-oct-2031 \
   SUPERSEDE ISSUED=21-jun-2010 ck=252 \
   SN=RK:2925-0:868730:325472 START=20-oct-2011 \
   AUTH={ snpslmd=(LK=CD4664E94E6462682FFD) }

INCREMENT Formality snpslmd 2013.03 31-dec-2013 2 \
   VENDOR_STRING=^1+S SUPERSEDE ISSUED=31-dec-2014 \
   ck=235 SN=RK:2593-0:868731:325472 START=20-oct-2011 \
   AUTH={ snpslmd=( LK=BD060419EAC935620CE5) \
   synopsysd=( LK=9D46A49985614122D3F6) }

INCREMENT Formality-E1 snpslmd 2013.03 31-dec-2013 2 \
   VENDOR_STRING=^1+S SUPERSEDE ISSUED=31-dec-2014 ck=46 \
   SN=RK:2593-0:868731:325472 START=20-oct-2011 \
   AUTH={ snpslmd=( LK=AD36F4F926AD783DC553) \
   synopsysd=( LK=9DB6D48931588E6D8556) }

INCREMENT PrimeTime snpslmd 2012.12 31-dec-2013 3 \
   VENDOR_STRING=^1+S SUPERSEDE ISSUED=31-dec-2014 \
   ck=4 SN=RK:2595-0:868738:325472 START=20-oct-2011\
   AUTH={ snpslmd=( LK=9D76D499C7EE3BAC14EF) }

INCREMENT hspice snpslmd 2013.03 31-dec-2013 2 \
   VENDOR_STRING=^1+S SUPERSEDE ISSUED=31-dec-2014
   ck=50 SN=RK:6719-0:868733:325472 START=20-oct-2011\
   AUTH={ snpslmd=( LK=5D86A4B97689DA46C550) \
   avantd=( LK=8DB6A4B9820FC454A318) }

INCREMENT hspiceva snpslmd 2013.03 31-dec-2013 2 \
   VENDOR_STRING=^1+S SUPERSEDE ISSUED=31-dec-2014 \
   ck=128 SN=RK:6719-0:868733:325472 START=20-oct-2011\
   AUTH={ snpslmd=( LK=3D76F4496C6D4D59F41F) \
   avantd=( LK=9DE6C4A90DD695C41E0C) }

INCREMENT metawaves snpslmd 2013.03 31-dec-2013 2 \
   VENDOR_STRING=^1+S SUPERSEDE ISSUED=31-dec-2014
   ck=247 SN=RK:6719-0:868733:325472 START=20-oct-2011\
   AUTH={ snpslmd=( LK=2D5694F9BBEAF2479700) \
   avantd=( LK=6D06641964A4A04CC5D0) }

INCREMENT synplify snpslmd 2013.03 31-dec-2013 2 \
   VENDOR_STRING=fpga SUPERSEDE ISSUED=01-jan-2010 ck=131 \
   SN=RK:4879-0:1067382:810881 START=20-oct-2011 AUTH={ snpslmd=( \
   LK=6D3CBBD68ED81E827B1F) \
   synplctyd=( LK=CD1C9BE681B64B965B1E SIGN="031B B750 0930 705B \
   9734 3B5A C3CB BFF5 ADA2 C565 8F00 EB94 76F4 9B29 D8E4 4C7A \
   E1FA 20C9 9154 7860 4D52") }
#----------------------------------------------------------
```

8

Maintaining Synopsys Common Licensing

This chapter describes how to maintain the SCL licensing software when you change the host system used as the license server, add new license servers (rehosting), or obtain new license key files.

This chapter includes the following sections:

Maintaining Synopsys Common Licensing

Over time, your licensing system and network can undergo changes. For example, you might change the machines you use to host your license servers or you might add servers. Most likely, you will change the number or kind of licenses you use. This section explains related tasks.

Here are the most common tasks you perform in maintaining your SCL licensing system:

- Adding monthly technology subscription licenses. When you request redemption of a monthly technology subscription license, you are sent an e-mail message containing that license. You can add this license to your license key file by using any text editor.
- Rehosting. Whenever you change the host machine used as your license server, you must “rehost” the system. This process entails obtaining a rehost form from your sales representative and using this form to communicate the licenses to be moved as well as the host IDs of the new server. Only licenses that are on maintenance (on support) can be rehosted.
- Retrieving license keys. When you retrieve your license keys for a site ID using SmartKeys, the license key file contains the full complement of active licenses for that site, including
  - Newly purchased term licenses
  - Previously issued and active term licenses
  - Previously redeemed and active monthly technology subscription licenses

  SmartKeys maintains the current state of your license key file and always returns the superset of active licenses, eliminating any expired or off-support licenses. You must entirely replace your old license key file with the new one. Only temporary licenses can be added to the license key file. Adding any other licenses voids the entire file.

Important:
Any time features are added or removed in the license key file, run the `sssverify` utility on the key file before restarting the license server to ensure license key file integrity.

SCL License Key File Verification Checklist

To avoid unpredictable license daemon behavior and license denials, follow these guidelines:

- Use the complete license file as received from Synopsys or as retrieved from the SmartKeys system. You must not add, delete or modify any INCREMENT lines within the license file. As new transactions occur (maintenance renewals, remixes, updates, rehosts, and so on), the entire license file is regenerated and again must be used without any modifications.
- If you receive temporary or monthly technology subscription license keys from Synopsys, you can append them to the existing revenue license file. These keys must have an issue date equal to or greater than the issue date of the on-support revenue keys with the same feature name (check the `ISSUED` field of the license key file).
- It is strongly recommended that you use a separate license server for temporary or monthly technology subscription license keys. This helps to ensure that these keys are not invalidated by subsequent revenue keys with a later issued date.

Validating New SCL Key Files

When you receive a new Synopsys license key file by e-mail or through the SmartKeys system, you must run the `sssverify` utility before using the new key file. Run the `sssverify` utility on the license file to identify any errors before starting the server. To use the file, enter

```
% scl_root/platform/bin sssverify /path/to/synopsys.lic
```

You must also use this utility whenever the license file is updated. The `sssverify` utility authenticates the integrity of the license file and detects any formatting errors in the file.

To start `sssverify`, enter

```
% sssverify license_file
```

If you receive a license error message, such as “License file integrity check failed!,” see Using sssverify to Troubleshoot a Synopsys License File.

Validating Existing or New SCL License Files

All regular (release) keys must have a valid SSS key. A missing or invalid SSS key makes the license file invalid and can result in unpredictable license daemon behavior.

As a precaution, check your SCL license debug log file for SSS errors.

For example, a SSS warning from a debug log file without a valid SSS key is shown here:

```
12:54:03 (snpslmd) WARNING: SSS errors.
12:54:03 (snpslmd) Use the sssverify utility to check the
      integrity of your license file.
12:54:03 (snpslmd) The license file should be used exactly
      as received from Synopsys.
```

Important:

> If the license server log file includes the previous error message, this indicates that the file is corrupt. You must rectify the error messages by invoking the sssverify utility before you proceed. If the problem is not corrected, the license server might behave erratically and users might see random license denials. For troubleshooting details about sssverify, see Using sssverify to Troubleshoot a Synopsys License File.

If you encounter this type of error in your file, retrieve the latest key file from SmartKeys at http://www.synopsys.com/smartkeys

Procedure to Update a Synopsys License File

Typically, when your current license file is about to expire, you receive a new license file from Synopsys. You can always retrieve your latest keys from SmartKeys (see Logging In to SmartKeys).

If you are replacing an existing license file that is currently running under SCL, use the following instructions that applies to your situation:

- Scenario 1 – New license file dates overlap the expiration dates of the existing, about-to-expire license file.
- Scenario 2 – New license file dates do not overlap the expiration dates of the existing, about-to-expire license file.

Scenario 1

In this scenario, the new license key file date overlaps with the existing, about-to-expire license file. For example, assume that your current license file expires on 1 June 2014, and the new license file has a start date of 31 May 2014.

To update your license key file on or after the start date of the new license file, follow these steps:

1. Make sure that you have the latest version of Synopsys Common Licensing (SCL) installed. To download the latest version of SCL, go to the SolvNet Download Center at https://solvnet.synopsys.com/DownloadCenter/dc/product.jsp

2. Verify the accuracy of the new license file by using the `sssverify` utility, which is included with SCL. Use the following syntax:

   ```
   % cd scl_root/platform/bin
   % sssverify /path/to/license/file
   ```

3. Rename the synopsys.lic file to the file name synopsys.lic.old.

4. Rename the new license file to the file name synopsys.lic.

5. Make sure that the host name, port number, and vendor daemon path information are correct. To ensure that no user jobs are disconnected from the license server, copy the SERVER and VENDOR lines from the old key file. The `VENDOR` path information can change, but only if you are upgrading to a new version of SCL.

6. Do not add any keys (INCREMENT lines) to the new license file unless they are temporary keys that are issued after the new license file was issued. (To determine whether they are temporary, you can compare the `ISSUED` date fields.)

7. Use the `lmdown` command to stop the current license daemon.

8. Make sure the `lmgrd` and `snpslmd` processes are no longer running. Also verify that all Synopsys lock files for `snpslmd`, `avantd`, `tmald`, `nassd` and so forth are removed from the /usr/tmp directory. For example, remove files such as `locksnpslmd`, `lockavantd`, `locktmald`, `locknassd`.

9. Restart the SCL license server by using the `lmgrd` command.

   Important: You should not use the `lmreread` command when you are replacing an entire keyfile.

10. Check the debug log file to make sure the server has started correctly.

Scenario 2

New license file date does not overlap with the existing, about-to-expire license file. In this scenario, your existing license file expires today and the new one does not start until tomorrow. For example, if your current license file expires on 30 June 2014 and the new license file is valid starting 2 July 2014. (To determine when the new keys start, check the START field.) In this situation, follow these steps:

1. Verify the accuracy of the new license file by using the `sssverify` utility, which is included with SCL. The syntax is the following:

```
% cd scl_root/platform/bin
% sssverify /path/to/license/file
```

2. Rename the synopsys.lic file to the file name synopsys.lic.old.
3. Rename the new license file to the file name synopsys.lic.
4. Make sure that the host name, port number, and vendor daemon path information are correct. This information must be the same as that for the synopsys.lic.old file. To ensure accuracy, copy the SERVER and VENDOR lines from the old key file.
5. Do not restart the server by using the `lmgrd` command. Do not use `lmreread`. The server automatically rereads the license file, which is renamed to the old license file, at 12:00 midnight the next day, when the keys from the old license file expire.
6. Verify that the keys started any time after midnight the next day.
7. Check the debug log file to make sure the server has started correctly.

License Expiration Alert

The `SCL_DISPLAY_EXPIRE` and `SCL_DAYS_EXPIRE` variables issue warning messages to notify license administrators of impending license feature expiration. The warnings are displayed in the debug log file.

SCL_DISPLAY_EXPIRE

The `SCL_DISPLAY_EXPIRE` variable allows you to set the frequency of your license expiration warning messages in minutes.

Syntax

```
setenv SCL_DISPLAY_EXPIRE  minutes
```

where *minutes* is the number of minutes between license expiration warnings.

SCL_DAYS_EXPIRE

The `SCL_DAYS_EXPIRE` variable allows you to specify how soon before the license expiration date the warning messages should start.

Syntax

```
setenv SCL_DAYS_EXPIRE  days
```

where *days* is the number of days before the license expiration that you want the warning messages to start.

Example

```
% setenv SCL_DISPLAY_EXPIRE 10
% setenv SCL_DAYS_EXPIRE 30
```

These settings cause warning messages to begin 30 days before expiration, at a frequency of 10-minute intervals.

The following example shows a warning message printed in the debug log file. The message shows the feature name, the feature key, and the expiration date.

```
15:29:58 (snpslmd) EXPIRATION WARNING: f3 07EB8A0EC08C Jun-25-2014
```

Note:

If you do not set a value for `days`, warning messages begin 14 days before expiration, but only when the server is started or reread.

9

Identifying and Resolving Problems

This chapter explains some common problems you might encounter in configuring and running SCL software, including problems in starting up the license server and problems encountered by users attempting to acquire licenses.

- Common Problems
- Mapping SCL Error Messages to FlexNet Error Codes
- Error Conditions: Possible Causes and Solutions
- Troubleshooting
- Known Issues

Common Problems

The two most common problems you might encounter in running SCL are the inability to start the license server and the inability to check out licenses. The following sections explore some of the reasons these problems might occur:

- Determining Why You Cannot Start the Server
- Diagnosing Why You Cannot Check Out a License

Determining Why You Cannot Start the Server

If you cannot start the license server, the most important step in debugging is to look at the debug log file. This file shows specific errors. It also shows the FlexNet version of the `lmgrd` and `snpslmd` utility, as well as the SCL version. The FlexNet version of `lmgrd` and `snpslmd` must be version 11.14.1.3 or later for all platforms. Make sure these versions are correct.

General things to look for in the license key file:

- Verify that the license file has read permissions.
- Verify the integrity of the license key file by using the `sssverify` utility. For more information about `sssverify`, see Checking the Integrity of the License File.
- Verify that the host ID specified on the SERVER line is correct and matches the results of `lmhostid`. (This number cannot be changed.)
- Verify that the SERVER name is correct and that you are logged into this server.
- Verify that the VENDOR line specifies the correct path to `snpslmd.`

For details on what to look for in the file, see Checking Requisite License Key File Content: SERVER and VENDOR Lines.

Other things to look for:

- Make sure you used the correct syntax to start `lmgrd` and that the license key file (synopsys.lic) is in the location specified.

  The basic syntax is

  ```
  lmgrd –c synopsys.lic –l debug.log
  ```
- Use the `lmdown` utility to stop previous instances of SCL (`lmgrd` and `snpslmd`).
- On Linux, use the `ps` command to verify that all instances of the `snpslmd` daemon have been stopped, as well as the parent `lmgrd` process.
- Make sure that you are not running any legacy daemons. For a list of legacy daemons, see Synopsys Legacy Vendor Daemons.

- On Linux, check the /usr/tmp directory and remove any lock files (lock*) for snpslmd and legacy daemons. See Table 1-5 for a list of Synopsys legacy daemons.

Diagnosing Why You Cannot Check Out a License

When you cannot check out a license for a supported feature, you can run diagnostic utilities such as lmstat or lmdiag to help identify the source of the problem. For information about these utilities, see Using the lmdiag and lmstat Utilities to Diagnose Problems When You Cannot Check Out a License.

If lmstat or lmdiag cannot contact the server, verify that the snpslmd daemon is running on the server node.

Next, you should ensure that there are no problems in the following areas:

- Verify that the license environment variables are set correctly on the client attempting to check out the license. Ensure that the license environment variable points to the server whose license key file contains the license for the required feature. For information about setting environment variables, see Describing the License Environment Variables.
- Use the sssverify utility to validate the integrity of the license file and detect any errors in the file. It is mandatory to use the complete combined vendor daemon license file as received from Synopsys. The license file should not be edited or merged with other license files. For more information about sssverify, see Verifying the Content and Integrity of the License Key File.
- Verify that you can connect to the server. Use any TCP-based command, such as telnet, to connect to the server. If the system hosting the server is down when the client first attempts to obtain a license, the reported message might indicate that no licenses are available.
- Verify that there are no network connectivity or communications problems between the client and the server. SCL general messages reporting that a license is not available could also indicate a network connection problem. The message would not identify this explicitly as the source of the problem, but you should check for the condition.

Mapping SCL Error Messages to FlexNet Error Codes

Table 9-1 contains some of the most commonly encountered error conditions. It lists the FlexNet error code for an error condition, the corresponding Synopsys product-specific licensing error message displayed to a user, and one or more notations indicating corresponding causes and solutions. Corresponding causes and solutions are explained separately from individual error messages because they might apply to many different error conditions and reported error messages.

For an explanation of corresponding causes and solutions indicated by the notations, see the Causes and Solutions sections.

Table 9-1 Mapping FlexNet Error Codes to SCL Error Messages and Causes

FlexNet error code and meaning	SCL product-specific equivalent error messages	Corresponding causes and solutions
-1 (LM_NOCONFFILE) Cannot find license key file. -2 (LM_BADFILE) Invalid license key file syntax. -30 (LM_NOREADLIC) Cannot read license key file.	SEC-2: Cannot open key file *filename*. or A licensing environment parameter has not been set correctly.	C12, C14 S13
-8 (LM_BADCODE) Invalid (inconsistent) license key. The license key and data for a feature do not match. This usually happens when a license file has been altered. -9 (LM_NOTTHISHOST) Invalid host. The host ID of this system does not match the host ID specified in the license file.	SEC-10: Software is not licensed for this machine.	C1, C3, C4, C5, C6 S1, S2, S3, S4
- 31 (LM_TOOEARLY) Feature start date is in the future.	SEC-11: Software is not yet enabled or has expired.	C3, C4 S2, S3

Table 9-1 Mapping FlexNet Error Codes to SCL Error Messages and Causes (Continued)

FlexNet error code and meaning	SCL product-specific equivalent error messages	Corresponding causes and solutions
-3 (LM_NOSERVER) No server for this feature. -15 (LM_CANTCONNECT) Can't connect to license server: - The lmgrd server has not been started yet. - The wrong port@host or license file is being used. - The port or host name in the license file has changed. -29 (LM_CANTFINDETHER) Cannot find Ethernet device.	SEC-12: Can't communicate with the license server.	C8, C9, C10, C11, C12 S8, S9, S10, S11 See What Happens When the License Server Goes Down.
-34 (LM_CLOCKBAD) Clock difference is too large between client and server.	SEC-13: The date/time difference between the client and the license server host is too great.	C16, S15
-11 (LM_BADDATE) Invalid date format in the license file. -13 (LM_NO_SERVER_IN_FILE) No SERVER lines in license file. -14 (LM_BADHOS) Cannot find SERVER host name in network database. The lookup for the host name on the SERVER line in the license key file failed. This often happens when NIS or DNS or the hosts file is incorrect. As a workaround, use the IP address instead of host name.	SEC-14: Key file syntax error: *@keyfile_name*.	C1, S1, S2

Table 9-1 Mapping FlexNet Error Codes to SCL Error Messages and Causes (Continued)

FlexNet error code and meaning	SCL product-specific equivalent error messages	Corresponding causes and solutions
-4 (LM_MAXUSERS) Licensed number of users already reached. -24 (LM_USERSQUEUED) Users are queued for this feature.	SEC-50: All *feature_name* licenses are in use.	C6, S6
-5 (LM_NOFEATURE) No such feature exists. -25 (LM_SERVERLONGGONE) License server does not support this version of this feature. -21 (LM_OLDVER) License file does not support this version.	SEC-51: This site is not licensed for *feature_name*.	C4, S2, S3, S6, S9
-26 (LM_TOOMANY) Request for more licenses than this feature supports.	SEC-52: Requested more licenses for *feature_name* than supported in the key file.	C3, S3
-38 (LM_FEATEXLUDE) User/host on EXCLUDE list for feature.	SEC-53: The end-user license options EXCLUDE you from using *feature_name*.	C7, S16
-39 (LM_FEATNOTINCLUDE) User/host not on INCLUDE list for this feature.	SEC-54: The end-user license options don’t INCLUDE you for using *feature_name*.	C7, S16
-23 (LM_BUSYNEWSERV) License server temporarily busy (new server connecting).	SEC-82: License server is busy, retrying.	C16

Table 9-1 Mapping FlexNet Error Codes to SCL Error Messages and Causes (Continued)

FlexNet error code and meaning	SCL product-specific equivalent error messages	Corresponding causes and solutions
-12 (LM_BADCOMM) Invalid returned data from license server. -16 (LM_CANTREAD) Cannot read data from license server. -17 (LM_CANTWRITE) Cannot write data to license server. -22 (LM_CHECKINBAD) Feature check in failure detected at license server.	SEC-85: Communication with the license server failed.	C9, C10, C11 S10, S11, S12, S13

Error Conditions: Possible Causes and Solutions

This section presents the most common causes of error conditions, the symptoms they present, and the ways to resolve the problems. Here is how this information is organized:

- Each possible cause is labeled with the letter C (for cause) followed by a number.
- Each possible solution is labeled with the letter S (for solution) followed by a number.

Causes and solutions are referred to by their labels from various entries documenting error messages in Table 9-1.

Causes

- C1. The INCREMENT line for the license key entry is corrupt or the INCREMENT line for the license has been modified, which invalidates the entire line, and thus, the license key file.
- C2. The license has expired.
- C3. The start date for the license has not yet occurred.
- C4. Licenses are not provided for the requested feature on any of the specified servers.
- C5. There are no licenses for the version of the feature requested.

- C6. The license daemon located one or more licenses for the requested feature on one of the specified license servers, but all licenses are in use.
- C7. The user requesting the feature has been excluded from access to a license for this feature or has not been explicitly granted access.
- C8. The VENDOR line in the license key file might include specification of an invalid path to the `snpslmd` daemon, or the VENDOR line specifies an options file and path without specifying the `options=` keyword. In this case, the single path specification on the VENDOR line is interpreted to be that of the `snpslmd` daemon.
- C9. The `lmgrd` and `snpslmd` daemon versions are incompatible. The version of the `lmgrd` daemon that is running is less than the version of the `snpslmd` daemon. If the `lmgrd` daemon is not equal to or greater than the version of the `snpslmd` daemon, results are unpredictable.
- C10. The `snpslmd` binary has been corrupted.
- C11. The tool (application program) is referencing the wrong license key file.
- C12. The license key file was modified; perhaps the host ID on the SERVER line was modified or an INCREMENT line was modified.
- C13. The license environment variable is set incorrectly.
- C14. The license key file is not in the specified location.
- C15. The permissions on the license key file render it unreadable by the SCL daemons.
- C16. The date or time value has been modified.
- C17. The license server is busy.

Solutions

- S1. Check the INCREMENT line for the license to ensure that it contains all the required fields and that all lines but the last are terminated with a continuation character.
- S2. If you are on support for this feature or have purchased a new license for the feature but have not yet retrieved the key, Downloading SCL Software and Obtaining License Key Files for details on how to obtain it.

  If you have not purchased a new license for the feature, but want to do so, contact your sales representative.
- S3. If the license is not yet active, you must wait until the license is available to use the feature, or contact your sales representative to have the license reissued.

- S4. Check to determine if other license servers not specified by the client contain the requested feature or the version of the requested feature. If so, reconfigure the client to refer to that license server.
- S5. Use `sssverify` to check the license key file containing the license for the feature to ensure that the line has not been modified or corrupted.

  If the license key entry for the feature is corrupt, obtain a new license key file from Synopsys. See Downloading SCL Software and Obtaining License Key Files.
- S6. Check to determine if another license server not specified by the client contains the requested feature or the version of the requested feature. If so, reconfigure the client to refer to that license server.
- S7. Check the VENDOR line in the license key file to ensure that if a path to `snpslmd` is specified, it is the correct one, and that if an options file path is specified, the `OPTIONS=` keyword is used.
- S8. Run the `lmdiag` utility to diagnose the source of the problem.
- S9. Verify that the tool is configured to reference the correct license key file.
- S10. Ensure that the `snpslmd` daemon is running.

  If you suspect that the `snpslmd` binary has been corrupted through a system failure or otherwise, replace the binary with the one in the original installation directory.
- S11. Obtain a new version of the license key file. For information about how to do this, see Downloading SCL Software and Obtaining License Key Files.
- S12. Set the license environment variables correctly to refer to the license key file for the target license server.
- S13. Contact the Synopsys Technical Support Center for assistance. For information about ways to contact the Center, see Customer Support.
- S14. Resolve the date or time conflict for the particular operating system.
- S15. Modify the options file to include the user (or remove the exclusion).

Troubleshooting

This section provides some useful, high-level approaches to troubleshooting common problems.

Determining Your Host ID by Using lmhostid

Use the FlexNet `lmhostid` command, which is included with SCL, to determine a license server's host ID for any supported platform. For example, execute the `lmhostid` or `lmutil lmhostid` at the system prompt. In the following example, the host ID of the machine, 3cd92bfe3ec is returned:

```
% lmutil lmhostid

lmutil - Copyright (c) 1989-2017 Flexera Software LLC. All Rights
Reserved.
The FlexNet host ID of this machine is "3cd92bfe3ec"
```

Note:

On Windows and Linux platforms, this command returns multiple values. You can choose any of these values which correspond to the Ethernet or MAC address. The 12-character Ethernet address that you submit for your license request must match a physical network adapter on your machine. In some cases your computer might report the presence of a wireless device or virtual network adapter. Do not use one of these addresses for your host ID; they might not be reliable and can change unpredictably.

Using sssverify to Troubleshoot a Synopsys License File

SCL uses Synopsys secure server (SSS) technology. It is important that any combined vendor daemon license files received from Synopsys be verified by running the `sssverify` tool. The failure to verify your license file can lead to license denials.

Use the following procedure to verify a license file:

1. Make sure you have the latest version of SCL. Go to the SolvNet Download Center at https://solvnet.synopsys.com/DownloadCenter for download instructions.
2. Make sure the license file host ID information is correct. Verify the host ID by using the `lmhostid` command. (SCL includes `lmhostid`.)
3. Examine the license file.
   - Make sure all comment lines start with a pound sign (#) and do not wrap.
   - All lines must start with a pound (#) sign or a keyword such as `SERVER`, `VENDOR`, `USE_SERVER`, `PACKAGE`, or `INCREMENT`.

  - INCREMENT lines can wrap only after a continuation character (\).
  - The USE_SERVER line must not be removed.

4. You must not add any keys (features or INCREMENT lines) to the license file, with the following exception:

   For a given host ID, unexpired temporary keys with an `ISSUED` date equal to or later than the `ISSUED` date for the release keys can be added to the license file. Both short-term temporary keys (keys issued for more less than 60 days) and long-term temporary keys (keys issued for more than 60 days) require an SSST security key.

5. Run the `sssverify` utility:

   ```
   % sssverify /path/to/synopsys.lic
   ```

   An example output from the `sssverify` utility follows:

   ```
   Report generated on 04-June-2018 (SCL_2018.06)
   -------------------------------------------------------
   Checking the integrity of the license file....
   Valid SSS feature found.
   Licensed to Synopsys Inc.
   Siteid: 1001, Server Hostid: 001f296516d8, Issued on: 6/04/2018
   License file integrity check PASSED!
   -------------------------------------------------------
   You may now USE this license file to start your license server.
   Please do not edit or manipulate the contents of this license file.
   ```

   As indicated in the message, using the license file is safe if there are no errors related to `SSS`.

   Note:
   If the license file contains only short-term temporary keys, you get this warning message: “The `SSS` feature is invalid.” (However, the license file integrity check still passes.) In this case, the warning can safely be ignored.

6. If you get an error message indicating that “License file integrity check failed,” download the latest license file from SmartKeys at https://solvnet.synopsys.com/SmartKeys, and run `sssverify` again.

Note:
If the `sssverify` utility reports that a license file is invalid, you must resolve the errors. If the problem is not corrected, the license server might behave erratically and users might see random license denials.

Note that all temporary license keys, including node-locked license keys, have the `SSST` feature in the license key file. In addition, all Monthly Technology Subscription License (MTSL) license keys (peak keys) include the `sss` feature. A single SSS feature validates all peak revenue keys and other types of revenue keys.

A customer might receive a license key file that has either SSS, SSST, or both features in it. Customers should not attempt to modify or delete the SSST feature from the temporary license file. If there are multiple SSST features in their license files, the SSST features should be left as they are in the license file.

As with all license keys, any license key files received from Synopsys should be verified by running the SCL `sssverify` tool. A failure to verify the license file could lead to license denials. If you get any `sssverify` errors or any SSS errors in the license server log file, the tool issues license checkout request denials until the problem is resolved. For additional information about `sssverify`, see Using sssverify to Troubleshoot a Synopsys License File.

If you have removed any features from the license file, the message in Example 9-1 is shown.

Example 9-1 Invalid sssverify Command and Report: Removed Features

```
% sssverify license_file
----------------------------------------------------------
Here is a sample output from running this utility:
Integrity check report for license file "license_file".
Report generated on 04-June-2018 (SCL_2018.06)
Checking the integrity of the license file...
Valid SSS feature found. INVALID license file fingerprint.
Siteid: 1001, Server Hostid: 001f296516d8, Issued on: 6/04/2018
3 features are missing from this file.
All revenue keys ("SN=RK:..." on feature line) have been excluded.
License file integrity check FAILED!
----------------------------------------------------------
This is an INVALID license file. You SHOULD NOT use this license file.
Please use the license file as received from Synopsys, Inc.
```

As indicated in the error message, the license file should not be used to start the license server.

If you have added any features to the license file, the message in Example 9-2 is shown.

Example 9-2 Invalid sssverify Command and Report: Added Features

```
% sssverify license_file
Integrity check report for license file “license_file”.
Report generated on 04-June-2018(SCL_2018.06)
----------------------------------------------------------
Checking the integrity of the license file...
Valid SSS feature found. INVALID license file fingerprint.
Siteid: 1001, Server Hostid: 001f296516d8, Issued on: 6/04/2018
You have added 3 new features to this file.
All revenue keys ("SN=RK:..." on feature line) have been excluded.
License file integrity check FAILED!
----------------------------------------------------------
This is an INVALID license file. You SHOULD NOT use this license file.
Please use the license file as received from Synopsys, Inc.
```

As indicated in the error message, the license file should not be used to start the license server.

If the license server log file has sss error messages, the report in Example 9-3 is shown.

Example 9-3 Invalid sssverify Report: sss Errors

```
      % sssverify license_file
12:54:03 (snpslmd) WARNING: SSS errors.
12:54:03 (snpslmd) Use the sssverify utility to check the
integrity of your license file.
12:54:03 (snpslmd) The license file should be used exactly
as received from Synopsys
```

Important:

You should run sssverify on both short-term (keys with a duration of less than 60 days) and long-term temporary keys (keys with a duration of 60 days). All temporary license keys, including node-locked license keys, include the SSST feature in the key file. In addition, all Monthly Technology Subscription License (MTSL) keys (peak keys) include the SSS feature, so a single SSS feature validates all types of revenue keys.

When you run the sssverify utility on a key file, and the file contains neither the SSS nor the SSST feature, the message shown in Example 9-4 should appear.

Example 9-4 Running the sssverify Utility on a Key File

```
=====================================================
Integrity check report for license file "test.lic".
Report generated on 04-June-2018 SCL 2018.06
----------------------------------------------------------
Checking the integrity of the license file...
No SSS or SSST features were found in the license file.
Siteid: 1001, Server Hostid: 001f296516d8, Issued on: 6/04/2018
All revenue keys ("SN=RK:..." on feature line) have been excluded.
License file integrity check FAILED!
----------------------------------------------------------
This is an INVALID license file. You SHOULD NOT use this license file.
Please use the license file as received from Synopsys, Inc.
=====================================================
```

If the license file contains only the SSS feature but not the SSST feature, the message shown in Example 9-5 should appear.

Example 9-5 License File Containing SSS Feature but Not SSST Feature

```
=====================================================
Integrity check report for license file "test.lic".
Report generated on 04-June-2018 SCL 2018.06
----------------------------------------------------------
Checking the integrity of the license file....
Valid SSS feature found.
Siteid: 1001, Server Hostid: 001f296516d8, Issued on: 6/04/2018
No SSST key found.
```

```
SSST key is required to enable temp keys ("SN=TK:..." on feature line).
License file integrity check FAILED!
```

To resolve the `sssverify` errors, download a new copy of the license key file from SmartKeys and run `sssverify` again.

If the license file invokes a vendor daemon that is not supported in the current platform, the warning messages shown in Example 9-6 should appear.

Example 9-6 License File Containing Vendor Daemon on an Unsupported OS Platform

```
Warning: Incompatible OS platform found. The vendor daemon <arcd> is not
supported on linux64 platform.

Warning: Please go to http://www.synopsys.com/licensing or refer to the
SCL_2018.06 Admin Guide for a list of supported OS platforms.
```

To resolve these warning messages, refer to the list of supported OS platforms in the section Supported Platforms and run `sssverify` again on the supported platforms.

If you are unable to resolve the error messages in the license file by invoking the `sssverify` utility, open a case with the Synopsys Support Center at:
http://www.synopsys.com/support

Alternatively, you can call (800) 245-8005 or send an email to support_center@synopsys.com. If you are located outside North America, contact your local Synopsys Support Center.

Determining Whether Your Tool Is Enabled for SCL

While starting the license server, you might encounter problems in which the sources are not obvious. In that case, it is a good idea to verify that the version of the tool you are running supports SCL. To determine if a tool supports SCL, use the `whatscl` utility, installed in the *scl_root*/*platform*/bin directory on Linux systems and the *scl_root*\windows\bin directory on Windows systems.

If the product is enabled for SCL the, `whatscl` utility returns the version of SCL incorporated into the tool. For example, enter

```
% cd /synopsys/scl/linux64/bin
% whatscl /synopsys/syn/2017.09-SP5/syn/linux64/syn/bin/common_shell_exec

Synopsys Corporate Licensing (SCL) Release: version SDK_11.6.8.3,
Build 20170907; Label: SDK_11.6.8.3; Built on Sep 7 2017 at 03:34:03
```

Note:
> The `whatscl` utility does not return information for tools built on legacy daemons, even if they are SCL compliant.

Finding the Physical Location of the License File

To obtain the physical location of the license file, enter

```
% lmstat -c  port@host
```

For example, if you are obtaining the license from 27020@server1, use the following command on any machine in the same network:

```
% lmstat -c 27020@server1
```

```
lmutil - Copyright (c) 1989-2017 Flexera Software LLC. All Rights
Reserved.
Flexible License Manager status on Tue 06/15/2018 09:57
License server status: 27020@server1
License file(s) on server1:
/synopsys/scl/synopsys.lic:
```

Finding a List of All Tools Users

To retrieve a list of users, use `lmstat` with the `-a` option. Enter

```
% lmstat -c port@host -a
```

or

```
% lmstat -c /path/to/synopsys.lic -a
```

Make sure you point to one license server or file at a time.

Checking the Daemon Versions for Compatibility

The version of the `lmgrd` daemon must be later than or equal to the version of the `snpslmd` daemon.

To check the version of `lmgrd`, enter

```
% lmgrd -v
```

To check the FlexNet version of `snpslmd`, enter the following FLEXlm command:

```
% snpslmd -v
```

Note:

> The FlexNet version is different than the SCL version. To determine the SCL version, use the `whatscl` utility.

System Monitoring Script

Use the `sysmon.sh` utility script included with SCL version 2018.06 to identify the system or configuration issues that result in licensing errors. The `sysmon.sh` executable file and `sysmon.README.txt` file are in included in the *scl_root*/examples directory.

When running the tool, this script collects a variety of information from the system and logs it into a file. The type of information collected includes the following:

- Hardware information (for example, number of CPUs, CPU speed, available RAM, and swap space)
- Network configuration (for example, TCP parameters and network interfaces)
- Network statistics (tcp and udp counters)

Usage

Run the script in one of the following ways:

```
sysmon.sh -p port -s -r loopInterval [-l logfile] [-c client_hostname]
```

or

```
sysmon.sh -p port [-l logfile]
```

Use the second command sequence to log the system state at any instant.

Here are the options:

`-p` *`port`*

Specifies the `snpslmd` daemon port.

`-c` *`client_hostname`*

Specifies host name of the client. Prints socket connections information between host and client.

`-s`

Prints socket summary only. No other information is collected.

`-r` *`loopInterval`*

Allows the script to run continuously, while periodically sampling the system state. The sampling value is in seconds.

`-l` *`logfile`*

Logs output to a user-defined file. The output of this file is saved under /tmp in the following format:

sysmon_DDMonYY:HHMM_$pid.log

Environment

Uses the `SNPSLMD_PATH` variable to locate FlexNet binary files. The binary files are used to determine the release versions for `lmgrd` and the Synopsys daemon. The path should refer to the root installation directory and should contain `lmutil`, `lmgrd`, `snpslmd`, and `whatscl` binary files.

Supported Platforms

The following platforms are currently supported: Solaris, Linux (all systems), and AIX.

Synopsys Statistical Debug Manager (SSDM)

Synopsys Statistical Debug Manager (SSDM) is a new tool that gathers license server related performance data. The tool captures real time performance data and other system statistics from the license server system which is used to identify and resolve license server related issues. It is also used to monitor system activities for resource planning and to prevent license server failures. The SSDM tool eventually replaces the current sysmon utility script.

The SSDM tool is invoked using the ssdm_admin tool, which is located in the *scl_root/platform/bin* directory.

Note:
Currently, the SSDM tool is only available for the linux64 platform.

The following information is collected by the SSDM tool for further analysis:

Static System Data

System information collected by the SSDM tool only one time:

- CPU information
- Host information
- OS information
- Memory information

Dynamic Runtime Data

Runtime information collected by SSDM at regular intervals:

- CPU usage data
- Overall network activity
- License server open files/sockets
- Memory usage of snpslmd

Usage

The SSDM tool is highly configurable. You can enable or disable any or all of the data being gathered by the SSDM tool using a XML configuration file located in the *scl_root*/*platform*/*bin* directory. You can also specify the polling interval at which the dynamic data is captured in the XML file.

Data collected by the SSDM tool is stored in a SQLite database file. The location of the database file is specified in the XML file. The name of the database file is ssdm_*hostname*.db. The file can be sent to Synopsys for further analysis if needed.

Steps to use the SSDM tool:

- The SSDM module is invoked by the same user that administers the license server.
- The SSDM module runs on the same host on where the license server is running.
- Set the LM_LICENSE_FILE/SNPSLMD_LICENSE_FILE to port@host.
- Start the SSDM module with the following command:

  ```
  % ssdm_admin start
  ```

- Stop the SSDM module with the following command:

  ```
  % ssdm_admin stop
  ```

Example

The license server is running on the host named host1. Login as a root user, or as the owner of the SCL tree.

To start the SSDM module,

```
# (bash shell)
% unset LM_LICENSE_FILE
% export SNPSLMD_LICENSE_FILE=27020@host1
% <scl_root>/platform/bin/ssdm_admin start
Server Response: SSDM Collect Module started with pid - 3459
```

To stop the SSDM module,

```
# (bash shell)
% unset LM_LICENSE_FILE
% export SNPSLMD_LICENSE_FILE=27020@host1
% <scl_root>/platform/bin/ssdm_admin stop
Server Response: Stopping SSDM collect module...
```

The current status of the SSDM module follows:

```
# (bash shell)
% unset LM_LICENSE_FILE
% export SNPSLMD_LICENSE_FILE=27020@host1
% <scl_root>/platform/bin/ssdm_admin status
Status: Running
Start Time: 10/20/2014 10:34:55
```

Using the Debug Log File to Troubleshoot License Issues

The log file contains useful information to help you debug problems. When you start the `lmgrd` daemon to bring up the license server, be sure that you redirect the output into a log file (debug log) where you can examine it. To do this, use the `lmgrd` command with the `-l` option. For example,

```
% lmgrd -c licensefile -l logfilename
```

You must have write access to the directory containing the log file. The log file (as well as the license key file and other SCL components) should be locally mounted.

Note:
You should have your debug log file available when you call the Synopsys Support Center.

Example 6-1 shows an example debug log file output. The debug log file contains the following information:

- Start-up information
- SCL and FlexNet version information

  The example log file has FlexNet version 11.14.1.3 and SCL version 2018.06.
- TCP port information

  In the example log file, the `lmgrd` license daemon uses TCP port 27020 and the `snpslmd` vendor daemon uses TCP port 60733.
- Security information

  In the example log file, SCL reports that it has found a valid `SSS` feature in the license key file.
- A list of incorporated daemons. SCL supports clients based on various daemons. A list of supported daemons incorporated into SCL is included in the log file. The list of incorporated daemons follows the “Serving features for the following vendor names” line.

- Informational messages

  In most cases, informational messages do not require intervention. These time-stamped messages identify license check ins (IN:) and check-outs (OUT:) as well as license denials (DENIED:). They identify the user, client system, and feature, as well as other information.

- Error messages

  These messages identify configuration problems that require intervention. Problems such as the wrong host ID or an invalid host name are among the issues reported in this category. For example, messages might report that the daemon cannot open the options file, that no VENDOR line in the file exists for a specified daemon, or the license key file is corrupt and, therefore, the daemon could not find licenses to serve. Common error messages are described in the following section.

Common Error Messages Reported in the Debug Log File

This section explains some of the more common error messages recorded in the debug log file. (The debug log reports messages from the `lmgrd` and `snpslmd` daemons.)

- SSS-Related Errors
- Other Messages and Resolutions

SSS-Related Errors

The errors in this section are related to the `SSS` feature. Generally, to resolve `SSS`-related errors, follow these steps:

- Do not add or remove keys from the license key file.
- Make sure you are running the latest version of SCL. The FlexNet version for both `lmgrd` and `snpslmd` should be version 11.14.1.3 or later.
- Retrieve a new license file from SmartKeys.
- Run the `sssverify` utility to verify the integrity of the license key file.

Specific SSS (or SSST) error messages follow:

“The SSS feature is missing. The SSS feature is invalid. This feature is required to enable other keys in your license file”

This message is reported under the following conditions:

- The SSS key is either missing or invalid, or the sssverify utility indicates that the INCREMENT lines have errors.

“Valid SSS feature found. Invalid License file fingerprint”

The SSS feature is valid but licenses have been added, removed, or altered.

“You have removed 2 features from this file,” or “You have added 2 new feature to this file”

This message is reported under the following condition:

- Features have been removed from or added to the license file.

“Run sssverify utility to check the integrity of your license file. You should use the license file exactly as received from Synopsys”

This message is reported under the following conditions:

- You are not using an authenticated Synopsys license file.
- You have removed or added features from the license file, which invalidates the license file.

“All revenue keys (“SN=RK...” on feature line) have been excluded...The server is in an unstable state”

This message is reported under the following conditions:

- The SSS key is either missing or invalid.
- SCL indicates that the INCREMENT lines have errors.
- You have removed or added features from the license file, which invalidates the license file.

“The following Temp Keys have been excluded...The server is in an unstable state”

This message is reported under the following conditions:

- The SSST or (SSS) key is either missing or invalid.
- SCL indicates that the INCREMENT lines have errors.

- You have removed or added features from the license file, which invalidates the license file.

Other Messages and Resolutions

“Not a valid server host name”

The license server checks the host name on the SERVER line to ensure that you are running the license server on the correct host machine. This message is reported if you used an alias for the host in the license key file. The host name on the SERVER line must match the name displayed when you issue the `hostname` command.

“Failed to open the TCP port number in the license”

For more information, see Imgrd Error Message: Failed to open the TCP port number in the license.

“Retrying socket bind (address already in use)”

One cause of this problem is that SCL is restarted but the operating system has not yet released the port. To resolve this issue, restart SCL with the `-reuseaddr` switch:

```
% lmgrd -c synopsys.lic -l debug.log -reuseaddr
```

Another cause might be that another application is using the TCP/IP port number specified on the SERVER line of the license key file.

To resolve this problem, shut down the license server by using the `lmdown` utility. Then change the TCP port number in the license key file, restart the `lmgrd` daemon, and check the log file again.

“Imgrd refused vendor name list. This daemon will only serve licenses for the primary vendor”

This message is reported when a version of `lmgrd` is used that is not compatible with the `snpslmd` version. It is recommended that you use the `lmgrd` and `snpslmd` daemon only from the same SCL installation directory.

“inconsistent authorization code”

This error indicates that the host ID is not correct or that the keys are corrupt. Verify that SCL (`lmgrd`) was started on the correct license server and that the host ID is correct. Compare the host ID specified on the SERVER line with the results of the `lmhostid` command. If the keys are in error, contact your sales representative.

“inconsistent encryption code”

This message might be reported when a legacy version of the `snpslmd` vendor daemon is started. Check the debug log file to verify that SCL version 2018.06 or later has started. Make sure that the correct `snpslmd` path is specified in the VENDOR line of the license key file.

“tcp_s is bad”

This error message is reported if you stop the previous instance of the license daemon and try to restart the daemon before the kernel releases the daemon port. The daemon reuses the port number—even if the kernel is temporarily holding the port.

This error occurs if all of the following conditions are present:

- The server is hosted on a Solaris operating system
- The VENDOR line has a `PORT=` keyword in the license file

To resolve this issue, you must use the `–reuseaddr` option while starting the license daemon to resolve this issue. For example, enter

```
% lmgrd –c license_file -l  logfile -reuseaddr
```

“Unsupported”

If the tool you are running attempts to check out a feature that is not in your license file, the license log file records an “Unsupported” message. This message is not an error message, but rather a message notifying you that your server does not support some features.

This message might also appear if your license administrator has updated the license file or license server after you started your Synopsys tool. If you receive an “Unsupported” message, but you are sure that the feature you are trying to check out is supported, restart the tool you are running.

Filtering Information From the Server Log File

The SCL installation directory contains a tool named watchlog.c that filters the contents of the license server log file for certain types of messages. The watchlog.c file contains the source code and is part of the SCL installation (scl_root/example/watchlog.c).

You can use `watchlog.c` to monitor the FlexNet `lmgrd` output log for specific events. When an event is found, you can run a script or program to take a specific action.

For example, you can monitor the log file for “DENIED,” then run a script that e-mails the license administrator with information about the denial.

Building and Running watchlog.c

Note:

SCL is shipped with the watchlog binary and a watchlog.conf example configuration file, which are supported on all Linux and Windows platforms.

Because the tool is contained in one file, the build is straightforward:

```
% cc -o watchlog watchlog.c
```

Run the tool from the piped output of `lmgrd` rather than specifying a log file for `lmgrd` (the `-l` option to `lmgrd` is not used):

```
% lmgrd -c synopsys.lic | watchlog watchlog.conf -o lmgrd.log &
```

Note that the command must be run in the background even though `lmgrd` normally backgrounds itself. This is because watchlog does not automatically run in the background.

Setting up watchlog.conf

The configuration file for `watchlog` (which can be named anything you like if you use the same name as when you run `watchlog`) is a tab-delimited file with the pound sign (#) as the first character denoting a comment line. The first column is the regular expression you want to look for in the log file. The second column is the action to take when that expression is matched. There are three actions that can be taken: drop the line from the log, execute a program, or send an email.

To drop a line from the log, the action is the single word "drop." Anything else in the action column is assumed to be a program to run, as shown in Example 9-7.

Example 9-7 Dropping a Line From a Log

```
#This is the site "watchlog" configuration file for server
#"myserver". We're going to filter out "bad handshake" messages
#and notify a sys-admin if we see DENIED or WARNING messages.
#We are also going to alert specific users when we see a
#license expiry message.
#
#Format
#------
#pattern <tab> Action <space> Arguments
#where pattern = strings separated by spaces
#
#Regular Expression Action
# ------------------ ------
[bB]ad [hH]andshake drop
DENIED adminmail dave
WARNING adminmail john
expires mailto="user1@company.com, user2@company.com"
EXPIRATION WARNING mailto="user1@company.com, user2@company.com"
```

You can use the following e-mail addresses:

user1@company.com

user2@company.com

You can insert multiple tabs between the two columns, allowing you to align the columns; however, you must have at least one tab between the regular expression and the action. Any spaces in the regular expression are considered part of the expression. In Example 9-7, the “bad handshake” message is dropped from the final output log. When a DENIED line is shown, the “adminmail” program (binary or script) with the parameter “dave” is executed. The first word in the action column is the program name. All other words are the program parameters. When the program is executed, `watchlog` puts the line that triggered the action as the last parameter to the program call.

When a `WARNING` appears, “adminmail” with the parameter “john” is executed.

“mailto” is a new internal action that can be used to send an e-mail alert based on matching patterns. In Example 9-7, when either “expires” or “EXPIRATION WARNING” occurs, an alert is sent to user1 and user2. The tool accumulates all the expiration warnings and sends a consolidated e-mail every 24 hours.

Watchlog Utility Support on the Windows Platform

In SCL version 11.10, the Windows platform supports the watchlog utility. The components include:

- watchlog.exe – This is the main 32-bit windows executable.
- sendemail.pl – This Perl script sends e-mail alerts using the SMTP server.
- pcre.dll – This file is used by the watchlog executable during runtime.
- watchlog.conf – This is the default configuration file used by the watchlog binary.

If your configuration file includes the e-mail alerts action, you must install Active Perl. To download Active Perl, go to this address:

http://www.activestate.com/activeperl/downloads

E-mail alerts are sent through the SMTP server. In the configuration file, you need to specify the SMTP server name, the e-mail account sending the e-mails alerts, and the e-mail account receiving the e-mail alerts. Here is an example of this action in the configuration file:

```
# Regular Expression    Action
# ------------------    ------
  expires               mailto="test2@synopsys.com"
\                        mailserver="mailhost.synopsys.com" \
                        mailfrom="test1@synopsys.com"
```

What Happens When the License Server Goes Down

When the license server connection is lost, active licensed tools go through reconnection phases. Tools requiring no new licenses do not exit. However, if the tool requests a new license while the server is down, the license cannot be issued, and the tool issues an SEC-12 message for the requested license. If the tool fails or exits at this point depends on the tool and its design.

Verifying the Content and Integrity of the License Key File

Entry errors and corruption of data in a license key file could be the source of various problems both in starting up the licensing system and acquiring licenses. (See Checking the Integrity of the License File.)

Checking Requisite License Key File Content: SERVER and VENDOR Lines

It is good practice to visually check the file for typographical errors and incorrectly specified information that you might have introduced in modifying the file. Do this if you encounter problems in starting up the `lmgrd` daemon or if users have trouble checking out licenses for supported features, even if you checked the license key file when you first installed it.

Check the SERVER and VENDOR lines for typographical and informational errors. Ensure that the server host name and the port number are correct and that the `snpslmd` daemon path name and options file, if specified, are correct. Check the USE_SERVER line for correct spelling of the keyword.

At a minimum, visually check these aspects of the license key file:

- The SERVER line. Make sure the SERVER line specifies the correct host name for the host ID. The host name is the first parameter; the host ID is the second one. Errors in this entry could cause startup failures. Ensure that the port number is correct.
- The VENDOR line. Make sure the VENDOR line correctly specifies the SCL daemon, `snpslmd`. If you modified this line to also specify a path to the daemon, verify that the path information is correct. If you specified a path to the options file, ensure that the `snpslmd` daemon itself can be found. Errors in this entry could cause startup failures.

Checking the Host ID

The host ID for the machine hosting the license server is used in the license key encryption process. The host ID shown on the SERVER line of the license key file must be that of the license server machine.

The `lmhostid` utility displays output similar to the following message:

```
lmhostid - Copyright (c) 1989-2017 Flexera Software LLC. All Rights
Reserved.
The FLEXnet host ID of this machine is “001f296516d8”
```

Using the lmdiag and lmstat Utilities to Diagnose Problems When You Cannot Check Out a License

You can use the `lmdiag` and `lmstat` utilities to diagnose the source of the problem when a user is unable to check out a license. You can also use these utilities to gather information about licenses currently in use and who is using them.

Using the lmdiag Utility

You can use the `lmdiag` utility to test whether a license can be checked out for a specific feature when a user is denied the license. The utility prints out information about the feature, reporting whether the checkout failed or succeeded. If the license checkout fails, the reason for the failure is reported. If the server is down, the error message reports that this is the cause.

Here is the syntax for the `lmdiag` utility:

```
lmdiag [-c license_file] [-n] [feature]
```

Option	Description
`-c` *`license_file`*	This is the path to the file containing the feature to be diagnosed.
`-n`	Executes the utility noninteractively. In this mode, `lmdiag` does not prompt you for input. (In this mode, extended connection diagnostics are not available.)
`feature`	Specifies the feature to be diagnosed. If no feature is specified, all features in the file are diagnosed.

The `lmdiag` utility first prints information about the license and then attempts to check out each license. If the checkout succeeds, the `lmdiag` utility indicates this. If the checkout fails, the `lmdiag` utility displays the reason for the failure. If the checkout fails because the `lmdiag` utility cannot connect to the license server, you should run “extended connection diagnostics.” For complete information, see the Flexera *FlexNet End Users Guide*.

Using lmstat

You can use the `lmstat` utility to monitor the status of all network licensing activities, including all licenses in use and all users of licenses for specific features.

Here is the syntax for the `lmstat` utility:

```
lmstat [-a] [-A] [-c license_file] [-f feature_name]
[-i feature] [-s [server] [-S vendor]
```

Option	Description
`-a`	Displays all information.
`-A`	Lists all active licenses.
`-c` *`license_file`*	Specifies the license file to be checked.
`-f` *`feature_name`*	Lists users of the specified feature.
`-i` *`feature`*	Displays information about the named feature. If no feature is specified, displays information about all features.
`-s` *`server`*	Displays status of all license files listed in `$VENDOR_LICENSE_FILE` or `$LM_LICENSE_FILE`. If no server is specified, displays information about all servers.
`-S` *`vendor`*	Lists all users of the specified vendor's features.

Getting Help

If you encounter problems that you cannot diagnose yourself, you can seek help by contacting Synopsys Customer Support. For information about how to contact customer support, see Customer Support.

When you call Synopsys customer support, have the following information at hand:

- The license key file or files referenced by the client
- Description of your licensing environment, for example, whether you are using three-server redundancy or multiple distributed license files
- The platform on which the license server is running
- Your site ID, as specified in the license key file
- The version number of the `snpslmd` daemon that you are running (See Determining Whether Your Tool Is Enabled for SCL for details on obtaining the version number.)

- The debug log file
- The output of `lmstat` utility

Diagnosing Problems With FlexNet

Some problems you encounter might originate in FlexNet. To obtain diagnostic information about FlexNet, set the `FLEXLM_DIAGNOSTICS` environment variable. When this variable is set, FlexNet produces diagnostic information if a license checkout fails.

For information about how to set this variable and interpret diagnostic information, see the *FlexNet End Users Guide.*

Known Issues

This section contains known issues and workarounds for SCL version 2018.06.

Licensing Checkout (Heartbeat) Errors

A heartbeat is a message and acknowledgement sequence between a Synopsys tool and a Synopsys license server. Heartbeats are initiated by the tool and inform the tool when a server goes offline.

If the tool that sends the heartbeat message does not get a timely response from the license server, this lack of response is considered a heartbeat failure. Multiple heartbeat failures can cause licensing failures. Eventually the tool might start queuing, go into a suspended or sleep mode, or terminate abnormally.

Minimizing Heartbeat Failures

To minimize heartbeat failures, it is suggested that you have a dedicated license server with recent AMD or Intel x86_64 hardware running the latest version of SCL software. This server should be configured with the following:

- AMD or Intel x86_64 hardware
- Supported version of Red Hat Enterprise Linux 7.x
- 2-Gbps Ethernet connection (minimum 1 Gbps)
- 2-Gbps network switch (10 Gbps is recommended)
- Minimum 4 GB of RAM (8 GB of RAM is recommended)
- Single (not triple redundant) servers

Redundant servers are not recommended because they are not robust in recovering from failures. Instead, use multiple single servers to achieve redundancy.

SCL License Server Out-of-File Descriptors on Linux Operating Systems

The following error message indicates that the server does not have file descriptors:

```
(lmgrd) Can't open /usr/tmp/.flexlm/lmgrdl.3343, errno: 24
```

If you get this message, increase the file descriptor in your SCL startup script, before starting lmgrd

In general, you should use a file descriptor setting of 8 times the number of licenses. Therefore, if you have a total quantity of 1,000 licenses, the descriptor setting should be 8192. For information how to set a file descriptor, see

https://www.redhat.com/archives/redhat-sysadmin-list/2006-October/msg00007.html

lmgrd Error Message: Failed to open the TCP port number in the license

If you get the “(lmgrd) Failed to open the TCP port number in the license” error message, stop all SCL processes and restart the server.

On Windows systems,

1. Stop any SCL processes such as `lmgrd` and `snpslmd`.
2. Restart `lmgrd` by using `lmtools`.

If this does not help, change the TCP port specified in the SERVER line of the license file (see Modifying the SERVER Line), or contact your local Support Center.

On Linux systems,

1. Stop any SCL processes such as `lmgrd` and `snpslmd`. (Use the `kill` or `kill -15` command only; do not use the `kill -9` command).
2. Restart `lmgrd` with the `–reuseaddr` switch:

   ```
   % lmgrd –c synopsys.lic  -l debug.log  -reuseaddr
   ```

   If this process fails, continue with step 3:
3. Make sure the /etc/hosts file has a `127.0.0.1 localhost` entry. Enter

   ```
   % cat /etc/hosts
   ```

```
127.0.0.1  localhost
```

4. Edit the /etc/nsswitch.conf file.

   Locate the hosts and networks lines and make sure the first entry after the colon refers to `files`. For example,

   ```
   hosts: files [NOTFOUND=continue] dns
   networks: files [NOTFOUND=continue] nis
   ```

5. Save any changes and repeat steps 1 and 2.

If this does not help, change the TCP port specified in the SERVER line of the license file. For more information, see Modifying the SERVER Line or contact your local Support Center.

A

Starting SCL on Windows Platforms With LMTOOLS

This appendix is intended to help you configure, start, and troubleshoot a FlexNet license server on Windows platforms using LMTOOLS. This appendix includes the following sections:

- General Information and Installation
- SCL Components

General Information and Installation

This section contains the following topics:

- SCL Components
- Starting SCL Automatically With LMTOOLS

SCL Components

The following components make up SCL licensing on Windows platforms:

- `lmgrd`, `snpslmd`, LMTOOLS, and other FlexNet binary files distributed with SCL
- A parallel port or USB dongle (although the Ethernet ID can also be used)
- The dongle software driver
- The license key file obtained from Synopsys

Starting SCL Automatically With LMTOOLS

Follow these steps to install and run SCL with LMTOOLS:

1. Install SCL 2018.06 or later using the default path. The default installation folder for the binary files and tools is

   C:\Synopsys\SCL\2018.06\win32\bin

2. Edit the license file. Change only the SERVER and VENDOR lines:

   ```
   SERVER hostname1 FLEXID=9-07B4ABE9 27020
   VENDOR snpslmd /path/to/snpslmd
   ```

   a. Change the SERVER line to include the PC’s host name, and customize the port number if needed. For example,

   ```
   SERVER my_server FLEXID=9-07B4ABE9 27010
   ```

   Note:
   For performance reasons, specifying ports 27000 through 27009 (FlexNet default ports) on the SERVER line is not recommended.

   b. Change the VENDOR line to reflect the path to the `snpslmd` vendor daemon. For example,

   ```
   VENDOR snpslmd C:\Synopsys\SCL\2018.06\win32\bin\snpslmd.exe
   ```

Save the file as

C:\Synopsys\SCL\2018.06\admin\license\synopsys.lic

Installing the software creates Start menu items under Start > All Programs > Synopsys > SCL 2018.06.

3. For license files that specify a dongle host ID (also called a FLEXID), make sure the dongle is properly connected to your system. Check to see if the dongle is recognized by invoking LMTOOLS from the start menu and clicking the System Settings tab. The dongle ID is displayed in the large FLEXID box on the lower left.

   If the dongle is not recognized, install the dongle driver included with SCL. For detailed information about using dongles, see Ordering FlexNet Dongles and Installing Dongle Drivers.

4. Configure the `lmgrd` utility to run as a service in Windows using the following steps:

   a. Invoke LMTOOLS from the Start Menu (Start > All Programs > Synopsys > SCL 2018.06 > LMTOOLS).

   b. On the Service/License file page, select Configuration using Services.

   c. Click the Config Services tab.

   d. Enter a Service Name (for example, Synopsys) and press the Tab key. The following three fields should be empty.

   e. Next to "Path to the lmgrd.exe file," click Browse and browse to the path of lmgrd.exe inside the SCL installation tree.

   f. Next to "Path to the license file," click Browse and browse to the path of synopsys.lic

   g. Next to "Path to the debug log file," enter this path:

      C:\Synopsys\SCL\2018.06\admin\logs\debug.log

   h. At the bottom of the Config Services page, select "Use Services," then select "Start Server at Power Up."

   i. Click Save Service, then click Yes to confirm.

5. Click the Start/Stop/Reread tab. Make sure your service name is highlighted. Click Start Server.

6. Make sure the server has started correctly by viewing the debug log file. (Click the Config Services tab, then click View Log.) Verify that there are no `SSS` or other errors.

7. (Optional) From within LMTOOLS, go to the Server Status tab and click Perform Status Enquiry. A list of available licenses should appear.

8. Set your license variable to the correct port@host value (27020@*my_server* in the step 2 example):
    a. In the Control Panel, click System > Advanced > Environment Variables.
    b. Under System Variables, click New.
    c. In the New System Variable box, next to Variable name, enter `SNPSLMD_LICENSE_FILE` or `LM_LICENSE_FILE`.
    d. Next to “Variable value,” enter the TCP port, followed by the @ symbol and the host name specified in the license key file. For more information about licensing variables, see Describing the License Environment Variables.
9. Invoke your application. If you encounter any errors, see Troubleshooting Tool Startup Problems.

Ordering FlexNet Dongles and Installing Dongle Drivers

A dongle-based license can be distinguished from a regular license based on the Ethernet or MAC address by the FLEXid prefix it contains. For example,

FLEXID=9-07B4ABE9

Dongle licenses can be normal networked licenses usable by anyone on the network, or they can be node-locked licenses that can be run only on the system with the MAC address or dongle ID specified. Networked licenses have the FLEXid prefix and the dongle ID specified on the SERVER line; node-locked licenses have a special `HOSTID=` field in the individual INCREMENT lines.

To be compatible with the USB dongle, the Synopsys tool must be built with FlexNet 7.2g or later. You can check the FLEXlm or FlexNet version of the application by using the `lmver` utility included with SCL.

This section includes the following topics:

- Ordering a FlexNet Dongle for Synopsys Tools
- Installing Dongle Drivers on Windows Systems

Ordering a FlexNet Dongle for Synopsys Tools

For supported Windows and Linux operating systems, Synopsys supports both FLEXID=9- and FLEXID=10- USB dongle types, which support both server-based and serverless licenses.

Before you order a dongle, make sure that your Synopsys tool supports a FlexNet dongle. To determine if you can obtain dongle-based licenses for your Synopsys tools, contact your sales representative.

To obtain a dongle for your Synopsys product, order it from Flexera Software by following these steps:

1. Send an e-mail to flexid@flexerasoftware.com.
2. Request a quote for the desired type of dongle (FLEXID=9- or FLEXID=10-).

   Important:

   Make sure to specify that you are a Synopsys customer; otherwise, Flexera Software will not process your order.
3. Place the order by following the procedure specified by Flexera Software.
4. When you receive the dongle, send the dongle host ID to your sales representative. This host ID, which is needed to create your license key file, is printed on the dongle. If this order is related to the rehost of an existing license key file, you must fill a rehost form.

Installing Dongle Drivers on Windows Systems

For the dongle ID to be seen by the `lmgrd` command, you must install the dongle software driver. The drivers supplied with the dongle from Flexera Software and those supplied with Synopsys software support the following dongle types:

Dongle Type	Dongle Prefix
FlexNet USB (HASP HL)	FLEXID=9-
FlexNet USB (WibuKey)	FLEXID=10-
Sentinel	SKEY=
RSoft	RSOFT=

SCL is not guaranteed to work with older versions of dongle drivers. The Sentinel SuperPro dongle is not supported on Windows platforms for server-based licenses. It is supported only for uncounted (server-less) licenses on Windows.

For best results,

1. Uninstall any old drivers.
2. In the Control Panel, open “Programs and Features.” Remove any old drivers, such as FLEXID Driver, Macrovision Driver, Globetrotter FLEXID Drivers, or Sentinel System Drivers.
3. Run the `FLEXidCleanupUtility`.
4. Install the new drivers.

   For more information, see the DONGLE_README.txt and FLEXnet_ID_Dongle_Driver.pdf files.
5. With the dongle firmly inserted into the USB port, shutdown and restart the system.
6. Open LMTOOLS and look for the dongle ID on the System Settings page. Make sure that the dongle ID is correctly specified in the FLEXID box in the lower-left of the window.

   If the dongle ID is not correctly specified, see Troubleshooting Dongle Problems.

Node-Locked Uncounted and Counted Licenses

A node-locked license has a hostid on any INCREMENT line that is node-locked to a particular host. The two types of node-locked licenses are uncounted and counted.

Uncounted Node-Locked License Keys

You do not need a license server to have an uncounted node-locked license key. You can simply retrieve your new license key from the SmartKeys Web site. The following example shows a serverless, node-locked license:

```
INCREMENT synplifydsp snpslmd 2018.06 31-dec-2018 uncounted \
   VENDOR_STRING=fpga,nl \
   HOSTID= FLEXID=9-07B4ABE9 \
   ISSUED=21-jun-2018 ck=159 SN=RK:4842-0:1067383:810881 \
   START=20-Oct-2011 AUTH={ snpslmd=( \
   LK=6D3CBBD68ED81E827B1F) \
   synplctyd=( LK=CD1C9BE681B64B965B1E SIGN="031B B750 0930 705B \
   9734 3B5A C3CB BFF5 ADA2 C565 8F00 EB94 76F4 9B29 D8E4 4C7A \
   E1FA 20C9 9154 7860 4D52") }
```

The license file is node-locked to a USB dongle (FLEXid = 9-07B4ABE9). Make sure that you install the dongle driver before attaching the dongle and invoking the tool.

Counted Node-Locked License Keys

You must have a license server to have a counted node-locked license key. The following example shows a counted node-locked license:

```
INCREMENT synplifydsp snpslmd 2016.06 31-dec-2016 2 \
   VENDOR_STRING=fpga,nl HOSTID= FLEXID=9-07B4ABE9 \
   ISSUED=21-jun-2013 ck=159 SN=RK:4842-0:1067383:810881 \
   START=20-Oct-2011 AUTH={ snpslmd=( \
   LK=6D3CBBD68ED81E827B1F) \
   synplctyd=( LK=CD1C9BE681B64B965B1E SIGN="031B B750 0930 705B \
   9734 3B5A C3CB BFF5 ADA2 C565 8F00 EB94 76F4 9B29 D8E4 4C7A \
   E1FA 20C9 9154 7860 4D52") }
```

The license file is node-locked to a USB dongle (FLEXid = 9-07B4ABE9). Make sure that you install the dongle driver before attaching the dongle and invoking the tool.

Troubleshooting Dongle Problems

This section outlines the steps to follow in using a dongle and provides tips for dealing with problems that might arise. For the dongle ID to be seen by FlexNet, a FLEXID software driver must be installed. The software driver is configured on Windows to start as a device driver.

If a dongle is not recognized, any of the following causes might be responsible:

- The wrong version of LMTOOLS has been invoked
- No dongle driver is installed
- The existing driver is obsolete
- A conflict with an old driver or another device exists
- An operating system problem exists or the dongle is defective

To check for these problems, do the following:

1. Start only the LMTOOLS utility provided with SCL 2018.06.
2. Click the System Setting tab and make sure the correct dongle ID shows in the FLEXID field. If the correct ID does not show, try the following:
   a. Make sure the USB dongle is attached directly to the system, not to a USB hub device.
   b. Remove and reinsert the dongle.

c. Uninstall any old drivers by using the Programs and Features Control Panel application. Old drivers have names such as Globetrotter FLEXID Driver, Macrovision FLEXID Driver, and Sentinel System Driver.

d. Restart the system.

e. Run the FLEXIDCleanupUtility cleanup utility provided with SCL.

f. Run FLEXIDInstaller or SentinelProtectionInstaller (included with SCL) to install the latest drivers. Make sure to specify the correct dongle prefixes.

g. Restart your machine and repeat steps 1 and 2. If the dongle driver is still not recognized, try these steps on another machine or contact Synopsys Support.

Troubleshooting SCL Startup Problems

The following topics are included:

- SCL Service Fails to Start Automatically on Windows Platforms
- Errors in the Debug Log File

SCL Service Fails to Start Automatically on Windows Platforms

In some cases, the SCL server does not start properly when the system is restarted, even if LMTOOLS is configured to start the SCL service automatically.

This usually occurs when the system tries to start SCL before the dongle driver is loaded. In this case, SCL will not start because it cannot find a valid host ID. The solution is to delay the startup of SCL.

To enable service delay for a particular service,

1. Start the Registry Editor:

   a. Click Start > Run and enter `Regedit`.

   b. Locate the registry entry for the Synopsys license service:

      HKEY_LOCAL_MACHINE\SOFTWARE\FLEXlm License
      \Manager*service_name*

      where *service_name* is the name of the license server service you configured in LMTOOLS.

2. Click Edit > New > String Value.

   a. Type `serviceDelay` and press Enter.

   b. Right-click the `serviceDelay` icon you created and click Modify.

c. Enter a value of 3 to delay SCL startup by three seconds, then click OK (the value can be 1 through 20 seconds).

You should now see the following information in the right pane of the Registry Editor:

Name	Type	Data
serviceDelay	REG_SZ	3

3. Exit the Registry Editor.

Errors in the Debug Log File

The debug log file should be checked every time the server is started or reread. (The log file is a text file that is in the location specified on the Config Services page of LMTOOLS.)

Here are common debug log file errors and their solutions.

Problem:

The debug.log file contains a “Failed to open the TCP port number” error message after starting the license server.

Solution:

In LMTOOLS, stop the license server. In Task manager, stop all instances of `lmgrd` and `snpslmd`. Restart the Synopsys SCL service. If this doesn’t resolve the problem, change the TCP port specified on the SERVER line of the license key file and repeat the enable service delay steps.

If following these steps still fails to resolve the issue, you might have a personal firewall running. See Troubleshooting Firewall Problems.

Problem:

`SSS` errors in debug logfile.

Solution:

Download the latest license key file from the SmartKeys Web site and run `sssverify`.

Problem:

“Error - lmgrd refused vendor name list. This daemon will only serve licenses for the primary vendor” appears in the `lmgrd` log file and licenses cannot be checked out.

Solution:

This issue is most often caused by running an earlier version of `snpslmd` and trying to use a later version of SCL.

Problem:

“Future file format” error message in `lmgrd` log file when starting the server.

Solution:

You are likely using an older version of `lmgrd` with the current version of `snpslmd`.

Check the debug log file and make sure that the FlexNet version for `lmgrd` and `snpslmd` is version 11.14.1.3 or later. Download the latest version of SCL and use the binary files from this version.

Troubleshooting Tool Startup Problems

In this section, the following topics are covered:

- General Troubleshooting
- Troubleshooting Firewall Problems
- The FlexNet License Finder Dialog Box

General Troubleshooting

If the tool fails to start, verify that SNPSLMD_LICENSE_FILE or LM_LICENSE_FILE has been set to the correct port@host value. For troubleshooting purposes, consider setting both variables.

Make sure the license server is running and that the feature is available (in LMTOOLS, click the Server Status tab, then click Perform Status Enquiry).

If the feature is available, make sure that the version is correct. For example, a `hspice` or `hspicewin` 2017.12 feature is required to enable HSPICE 2017.12 or earlier unless you have an on-support key. If the version is not correct, retrieve the latest license key file from the SmartKeys Web site at https://solvnet.synopsys.com/SmartKeys

All tool requests should be logged with an appropriate checkout message.

If the request cannot be granted, you should see an error. Check the debug log file for `SSS` errors, `DENIED` errors, or `UNSUPPORTED` messages.

Note:

An UNSUPPORTED message is not necessarily an error. This message means that the tool requested a license that is not available in the license key file. Often, the tool looks for an alternate license.

Troubleshooting Firewall Problems

Firewalls and internet security programs (for example, Norton Internet Security) can block TCP ports and keep .exe files from running.

- If you have a personal firewall installed, make sure that lmgrd.exe and snpslmd.exe have TCP access privileges.
- If you have a network firewall, modify the license key file to assign the snpslmd daemon a port at the end of the VENDOR line (this is normally determined randomly.) Then open both the lmgrd and snpslmd ports. For example,

```
SERVER my_server 0003DC78E712 27020
VENDOR snpslmd C:\Synopsys\SCL_2018.06\admin\license\synopsys.lic
port=27021
```

The FlexNet License Finder Dialog Box

If no valid variable or feature (key) is available, or the specified server cannot be contacted, certain Windows tools display the FlexNet License Finder dialog box and request that you specify a license file or server name.

Assuming the requested feature is available, you might have to enter the port@host name multiple times until the license request is satisfied.

To resolve this problem, follow these steps:

1. Make sure the license server is running and available.
2. From within LMTOOLS, click the Server Status tab, then click Perform Status Enquiry. A list of available licenses should appear. If none do, click the Config Services tab, then click View Log for error information.
3. If the license is available, try setting SNPSLMD_LICENSE_FILE rather than LM_LICENSE_FILE (or set both values). If this fails, modify the license key file and change the lmgrd port (specified at the end of the SERVER line) to something outside the range of 27020-27009, such as 27020. Exit the tool and restart it.
4. If the previous step does not help, modify the license key file to assign the snpslmd daemon port to a value outside the range of 27020-27009. For example, add port=28001 to the end of the VENDOR line. Stop and restart the server.

Open the Control Panel and select System > Environment variables. Set `SNPSLMD_LICENSE_FILE` to the `snpslmd` port (for example, 27021) instead of to the `lmgrd` port.

Exit the tool and restart it.

B

SCL Command-Line Interface

This appendix describes the most commonly used SCL shell (sclsh) commands to test and validate the license server. This appendix includes the following sections:

- Getting Help on Commands
- Validating a License Key on the License Server
- Testing the License Server for Performance

Getting Help on Commands

SCL provides two levels of help information: a list of commands and command usage help.

Listing Commands

The `help` command lists the names of all sclsh commands. To display a list of all commands, enter `help` at the sclsh prompt.

```
% (sclsh) help
```

The following list appears:

```
ci            co            exit          feat_count    feat_list

get_srv_ver   help          init          lmstatus      manymcoci

mci           mco           mcoci         path          quit

sclpath       scl_version   server_debug  server_info   setenv

srvconn       status        usage         userlist      version
```

Getting Command Usage Help

To get help about a specific sclsh command, enter the command name and the `-help` option.

Syntax

```
command_name -help
```

Example

To get help about the `co` command, enter

```
% sclsh> co -help
```

The command usage `help` displays the command’s options and arguments.

Abbreviating Commands and Options

You can abbreviate `sclsh` command names and options to the shortest unambiguous (unique) string. For example, you can abbreviate the `release` command to `rel`. You can shorten an option name if the abbreviation is unique to the command. For example, to abbreviate the `coverage` command to `cov`, enter

```
%  sclsh> set coverage cov
%  sclsh> cov
```

Command abbreviation is meant as an interactive convenience. Do not use command or option abbreviations in script files because script files are susceptible to command changes in subsequent versions of the application. Such changes can cause abbreviations to become ambiguous.

Arguments

Many `sclsh` commands have required or optional arguments. These arguments allow you to further define, limit, or expand the scope of the command's operation.

- Arguments that do not begin with a hyphen (-) are positional arguments. They must be entered in a specific order relative to each other.
- Arguments that begin with a hyphen are nonpositional arguments. They can be entered in any order and can be intermingled with positional arguments. The names of nonpositional arguments can be abbreviated to the minimum number of characters necessary to distinguish the nonpositional arguments from the other arguments.
- Arguments are enclosed in square brackets ([]).

If you omit a required argument, an error message and a usage statement appear.

Comment Character

The sclsh command language uses the pound sign (#) to start a comment. The comment can start anywhere on a line. It ends with the end of the line.

Validating a License Key on the License Server

Use the following command sequence to check out a particular version of a key from the license server:

```
% sclsh
sclsh> init #Initialize the licensing module.
sclsh> co key_name [count] [version] [qflag] #Check out a
       license from the license server.
sclsh> free  #Free the licensing resources.
sclsh> ci key_name  #Return the license back to the license
       server.
sclsh> quit #Exit the application.
```

Use the `co` command to check out one or more keys of the specified version.

Syntax

`co` *`key_name`* `[`*`count`*`] [`*`version`*`] [`*`qflag`*`]`

These are the options:

`count`

The number of licenses to check out (must be > 0).

`version`

The version of the key in year, month format (YYYY.MM).

`qflag`

Check out option flag `q|w|l|n` where the flags are equivalent to the following options:

`q` — `LM_CO_QUEUE`, queue for a license if all the keys are consumed.

`w` — `LM_CO_WAIT`, queue for a license if all the keys are consumed.

`l, t – LM_CO_LOCALTEST`, verify whether a valid key is present, but do not check out.

`n – LM_CO_NOWAIT`, do not queue for a license if all the keys are consumed.

Testing the License Server for Performance

Two types of commands are available to test the license server performance.

- Commands to Validate the Socket Connection Limit
- Commands to Stress Test the License Server With Multiple License Requests

Note:

Do not use these commands on a production license server.

Commands to Validate the Socket Connection Limit

Use the multiple check out (`mco`) and multiple check in (`mci`) commands to establish or create multiple socket connections to the license server. Each socket connection checks out a single license from the license server and keeps the license until the `mci` command is invoked.

mco Command

Use the `mco` command to test the number of socket connections your license server can accept. This command does not stress test the license server. Each license is checked out through a separate socket connection. The licenses remain checked out from the server with the help of each socket connection until you release them by invoking `mci`.

The syntax is

```
mco key_name [count] [version] [qflag]
```

mci Command

Check in the license that was checked out by the `mco` command.

You can also use the secondary daemon prefix to check out the license from the combined vendor daemon secondary daemons.

The syntax is

```
[secondary_daemon] mco key_name [count] [version] [qflag]
[secondary_daemon] mci
```

Here is an example output of the `mco` and `mci` commands:

```
% sclsh> mco Design-Compiler 5 2018.06

Check-out Design-Compiler key checking out Design-Compiler
  (v2013.12) license #1...done checking out Design-Compiler
  (v2013.12) license #2...done checking out Design-Compiler
  (v2013.12) license #3...done checking out Design-Compiler
  (v2013.12) license #4...done checking out Design-Compiler
  (v2013.12) license #5...done

sclsh> mci Design-Compiler  #Check in all the Design-Compiler
       keys checking in ALL features...done
sclsh> quit  #Quit the application.
```

Commands to Stress Test the License Server With Multiple License Requests

Two commands are available to stress test the license server or bombard the license server with multiple requests: `mcoci` and `manymcoci`.

mcoci Command

The multiple check out and check in (`mcoci`) command checks out and checks in a license key in a continuous loop with the help of a single-socket connection (in the case of `mco`, the license is checked out only one time). You can specify that multiple socket connections be created so that check out and check in activities occur with each socket connection.

Syntax

```
[secondary-daemon] mcoci key_name [repetitions] [count]
  [-delay seconds]
```

The options are the following:

secondary-daemon

Indicates the secondary daemon name used for check out and check in.

key_name

Indicates the key name to check out or check in.

repetitions

Indicates the number of times to execute the check out/check in loop.

count

Indicates the number of keys to be checked out via each socket connection.

`-delay` *seconds*

Indicates the number of seconds to wait between check out and check in. The `mcoci` command creates multiple socket connections. The license key is checked out and checked in on each connection, in a `continue` loop. This happens as part of a single process. However, in a production environment where you have many computers in a farm, multiple processes interact with the license server. To simulate this environment, you must run multiple sessions of the `sclsh` utility with the mcoci command. To help achieve this goal, you can invoke the `manymcoci` command.

manymcoci Command

The many multiple check out and check in (`manymcoci)` command creates multiple `sclsh` processes. Each process creates multiple socket connections to the license server, and with the help of each socket connection, check out and check in of the license key can be performed in a loop. This simulates a compute farm environment. Make sure that you have large quantities of license keys available to test this scenario.

When you invoke the `manymcoci` command, a command prompt is provided that allows you to check the status of the processes invoked by the command (by invoking the `stat` command on the prompt). It also allows you to kill all the child processes and quit the application. After the `manymcoci` command completes its execution, it prints the status of all the child processes and the check out and check in status.

Syntax

```
[secondary-daemon] manymcoci feature [-quiet] [-delay seconds]
  [-children integer]
  [-childjobs integer] [-childloops integer]
```

The options are the following:

`secondary-daemon`

The vendor daemon to be used for check out or check in.

`key_name`

Specifies the feature to check out or check in.

`-quiet`

Disables the control prompt during the run (useful when using a `sclsh` input file).

`-delay seconds`

The number of seconds to wait between each check out and check in.

`-children` *`integer`*

The number of child processes to create. The value passed to the child process `mcoci` delay switch.

`-childjobs` *`integer`*

The number of socket connections the child processes should create for each check out and check in loop. The value is passed to the child process `mcoci` job-count parameter.

`-childloops` *`integer`*

The number of loops the child process should execute for the `check-out` or `check-in` command. The value is passed to the child process `mcoci` repetitions parameter.

When you are running `manymcoci`, the following commands are available at the command prompt:

`help`

Prints the available commands for the subprompt.

`stat`

Prints the current status of the child processes.

Here is an example of a `manymcoci` set of commands and the output report:

```
% sclsh> manymcoci Design-Compiler -delay 2-children 5 -childjobs 5 \
      -childloops 50
>>help  #A new prompt is displayed: the commands supported are stat,
      #help,and quit.
Enter one of the following commands: help, stat, quit
>>stat  #Shows the status of all the child processes.
```

Table B-1

ID	state	total	okays	fails	loop	error	stat
1	running	5	5	0	1		
2	running	5	5	0	1		
3	running	5	5	0	1		
4	running	5	5	0	1		
5	running	5	5	0	1		

When the output report is complete, the final status of all processes is printed to your screen before the main `sclsh` prompt reappears.

Command to Test the Connection to the License Servers

The `srvconn` command tests the connection to the license server for checkout operations. This command simulates multiple checkout requests, tracks the time it takes to complete the result, and reports whether each request was successful.

The `srvconn` command sends the requests in a loop for a specified time. Specify the time by using the `-minutes` or `-hours` options. Specify the delay between two successive requests with the `-ping` option.

To report the checkout requests in a log file use the `-log` option. The log contains a timestamp of the time taken to complete each request and reports any errors. Each line in the log corresponds to a single request with the values separated by tabs. This information can be imported into a spreadsheet.

Syntax

```
srvconn -ping seconds [-minutes num_of_minutes | -hours num_of_hours]
[-log file_path][-noprompt][-nosocket]
```

The options are the following:

-ping *seconds*

Indicates the delay in seconds between successive LM_A_LOCALTEST requests.

-minutes *num_of_minutes*

Indicates the time in minutes that the command runs. This option is ignored if the `-hours` option is specified first.

`-hours` *num_of_hours*

Indicates the time in hours that the command runs. This option is ignored if the -`minutes` option is specified first.

`-log` *file_path*

Indicates the file name and path for the log file with the checkout requests. This option is not required.

`-noprompt`

Indicates that the srvconn prompt is not given back. You are not able to run the `stat` or `quit` commands. Srvconn runs for a specified amount of time by blocking the client session. This option can be used with the `-log` option.

`-nosocket`

Disables monitoring open sockets on the server.

Here is an example of the log file:

```
Req No.    Timestamp             Runtime     Error    Sockets
1          04/27/2015 14:56:01    2.70         0        607
2          04/27/2015 14:56:06    2.74         0        611
3          04/27/2015 14:56:11    4.26         0        606
4          04/27/2015 14:56:16    2.87         0        598
```

When you are running `srvconn`, the following commands are available at the command prompt:

`stat`

Prints the current status. This indicates the number of requests sent to the server, elapsed time, number of failures, and maximum and minimum times for the checkout.

`help`

Prints help for available commands.

`quit` or `exit`

Stops the `srvconn` command.

An example of a `srvconn` set of commands and the output report follows:

```
% sclsh> srvconn -ping 5 -minutes 1 -log my_svr_connections.log

Testing license server connection to 26585@swe-opf-9
Type 'stat' to see the current status of the command in execution
Type 'quit' to exit the command
% sclsh> stat
Number of requests sent [6] in last [0.43] minutes with delay [5] seconds
Number of requests failed: [0]
Overall checkout time - Maximum Time [1.05] milliseconds, \
   Minimum time: [0.67] milliseconds

srvcon Summary:
Number of requests sent [12] in last [1.00] minutes with \
   delay [5] seconds
Number of requests failed: [0]
Overall checkout time - Maximum Time [1.05] milliseconds, \
   Minimum time: [0.66] milliseconds

% sclsh> quit
```

C

SCL License Key File Sort Utility

This appendix describes the Synopsys Key File sort (`sclsort`) utility. This appendix includes the following sections:

- Features
- Usage
- Examples

The Synopsys Key File sort (`sclsort`) utility is command-line based and can be used to reorder the Synopsys license key file that you retrieve from the SmartKeys Web site. This utility reorders the license keys in the license key file, based on several fields in the INCREMENT line.

Features

The `sclsort` utility has the following features:

- The `sclsort` utility writes the reordered license keys to a new license key file; it does not overwrite the existing license key file.
- All comment lines before the SERVER line are retained in the new license key file. (Note that comment lines are the ones beginning with #.)
- No changes are made to any of the individual INCREMENT lines.
- Any changes made by this tool to the license key file does not affect the output of other utilities, such as the `sssverify` utility and product-to-feature mapping.
- After a new license key file has been generated, you can run the `sssverify` utility to validate the license key file.

The `sclsort` utility can reorder the keys in a license key file, based on the contents of the following fields:

- Key name (`feat`)
- Expiration date (`expdate`)
- Start date (`startdate`)
- Version number (`version`)
- Issued date (`issuedate`)
- Number of keys (`numkeys`)

By default, the license key is sorted based on the license key name. The user has an option of specifying a max of two fields for sorting. If there are any expired keys in the key file, they are moved to the end of the key file, irrespective of sorting order.

This utility can be used with both legacy and CVD-formatted key files as well as node-locked license key files and floating license key files.

If you have revenue and temporary keys in the same license key file, the revenue key files are sorted separately from the temporary key files. For temporary keys, the block of temporary key files with the associated `SSST` key is sorted separately. If you have multiple blocks of temporary keys, each with its own `SSST` key, each block of keys is sorted separately. Every `SSST` key block can be identified by its transaction ID.

PACKAGE keys: The INCREMENT line that corresponds to a PACKAGE key is sorted along with the other INCREMENT keys. The PACKAGE key can be found before the first INCREMENT line associated with the PACKAGE.

Usage

The `sclsort` utility is located in the *scl_root*/*platform*/bin/ directory, as shown in the following syntax:

```
%  sclsort
[-h|-help]
-p sortfield [:a|:d]
[ -s sortfield [:a|:d]]
-i infile
[ -o outfile]
```

The `sclsort` utility has the following options:

`-h|-help` Prints the help message

`-p` Specifies the primary sort field with sortfield. If you do not use this option, the license key file is sorted based on key names. You can continue to sort each subsequent field in either ascending or descending order by specifying either - `:a` or `:d`, respectively. The default sort order is descending, except for when you use the key name.

`-s` Specifies the secondary sort field with sortfield. That is, after the file is sorted on the primary sort field, it is sorted again on this sort value. You can continue to sort each field in either ascending or descending order by specifying either - `:a` or `:d`, respectively. The default sort order is descending, except for when you use the key name. If you do not use the secondary sort field, the license key file is sorted only on the primary sort field.

`-i` Specifies the input license key file.

`-o` Specifies the output license key file. The default is stdout.

The `sclsort` utility supports the following sort fields:

`feat` Specifies to sort by key name

`expdate` Specifies to sort by expiration dates

`startdate` Specifies to sort by start date

`issuedate`	Specifies to sort by issue date
`numkeys`	Specifies to sort by the number of keys
`version`	Specifies to sort by the version number

Examples

The following subsections show various uses of the `sclsort` utility.

Using a Feature Name in Descending Order

The following example uses the input file in.lic:

```
INCREMENT certify snpslmd 2012.12 30-dec-2014 1 \
   VENDOR_STRING=altera,xilinx \
   ISSUED=18-jun-2009 ck=222 SN=TK:255682:464396 \
   START=18-jun-2009 AUTH={ synplctyd=( SIGN="0205 8DC3 BD05 CCBD \
   28A7 6940 D88A 21EA 09EE 95EF 4B03 E74F EAAA 4165 BCD6 DF4E \
   0CD1 DDE0 C9CC 7D50 56D0") }

INCREMENT spice snpslmd 2012.12 30-dec-2014 1 \
   VENDOR_STRING=^1+S SUPERSEDE ISSUED=20-apr-2007 ck=184 \
   SN=TK:5096-0:871303:843082 START=05-mar-2007 \
   AUTH={ snpslmd=( LK=71C3F3F78009) \
   sandwork=( SIGN=E3C45AB6E84E) }

INCREMENT APSolar snpslmd 2012.12 30-dec-2014 1 \
   VENDOR_STRING=^1+S SUPERSEDE ISSUED=20-apr-2007 ck=173 \
   SN=TK:5096-0:871303:843082 START=05-mar-2007 \
   AUTH={ snpslmd=( LK=4D45D90FAD6B) \
   avantd=( LK=A6F2F854633C) }
```

This example shows the sclsort utility sorting a license key file by using a feature name in descending order.

```
% sclsort  -i in.lic -o out.lic -s feat:d
```

This example prints the following sorted license key file to the file name out.lic.

```
INCREMENT spice snpslmd 2012.12 30-dec-2014 1 \
VENDOR_STRING=^1+S SUPERSEDE ISSUED=20-apr-2007 ck=184 \
SN=TK:5096-0:871303:843082 START=05-mar-2007 \
AUTH={ snpslmd=( LK=71C3F3F78009) \
sandwork=( SIGN=E3C45AB6E84E) }

INCREMENT certify snpslmd 2012.12 30-dec-2014 1 \
VENDOR_STRING=altera,xilinx \
ISSUED=18-jun-2009 ck=222 SN=TK:255682:464396 \
START=18-jun-2009 AUTH={ synplctyd=( SIGN="0205 8DC3 BD05 CCBD \
28A7 6940 D88A 21EA 09EE 95EF 4B03 E74F EAAA 4165 BCD6 DF4E \
0CD1 DDE0 C9CC 7D50 56D0") }

INCREMENT APSolar snpslmd 2010.12 30-dec-2014 1 \
VENDOR_STRING=^1+S SUPERSEDE ISSUED=20-apr-2007 ck=173 \
SN=TK:5096-0:871303:843082 START=05-mar-2007 \
AUTH={ snpslmd=( LK=4D45D90FAD6B) \
avantd=( LK=A6F2F854633C) }
```

Using the Expiration Date in Descending Order

The following example uses the input file in.lic:

```
INCREMENT APSolar snpslmd 2012.12 30-jun-2014 1 \
   VENDOR_STRING=^1+S SUPERSEDE ISSUED=20-apr-2007 ck=173 \
   SN=TK:5096-0:871303:843082 START=05-mar-2007 \
   AUTH={ snpslmd=( LK=4D45D90FAD6B) \
   avantd=( LK=A6F2F854633C) }

INCREMENT certify snpslmd 2012.12 30-dec-2014 1 \
   VENDOR_STRING=altera,xilinx \
   ISSUED=18-jun-2009 ck=222 SN=TK:255682:464396 \
   START=18-jun-2009 AUTH={ synplctyd=( SIGN="0205 8DC3 BD05 CCBD \
   28A7 6940 D88A 21EA 09EE 95EF 4B03 E74F EAAA 4165 BCD6 DF4E \
   0CD1 DDE0 C9CC 7D50 56D0") }

INCREMENT APSolar snpslmd 2012.12 30-dec-2014 1 \
   VENDOR_STRING=^1+S SUPERSEDE ISSUED=20-apr-2007 ck=173 \
   SN=TK:5096-0:871303:843082 START=05-mar-2007 \
   AUTH={ snpslmd=( LK=4D45D90FAD6B) \
   avantd=( LK=A6F2F854633C) }

INCREMENT spice snpslmd 2012.12 30-dec-2014 1 \
      VENDOR_STRING=^1+S SUPERSEDE ISSUED=20-apr-2007 ck=184 \
      SN=TK:5096-0:871303:843082 START=05-mar-2007 \
      AUTH={ snpslmd=( LK=71C3F3F78009) \
      sandwork=( SIGN=E3C45AB6E84E) }
```

This example shows the sclsort utility sorting a license key file by using the expiration date in descending order.

```
% sclsort -p expdate:d –i in.lic -o out.lic
```

This example prints the following sorted license key file to the file name out.lic.

```
INCREMENT APSolar snpslmd 2012.12 30-dec-2014 1 \
VENDOR_STRING=^1+S SUPERSEDE ISSUED=20-apr-2007 ck=173 \
SN=TK:5096-0:871303:843082 START=05-mar-2007 \
AUTH={ snpslmd=( LK=4D45D90FAD6B) \
avantd=( LK=A6F2F854633C) }
```

```
INCREMENT APSolar snpslmd 2012.12 30-jun-2014 1 \
VENDOR_STRING=^1+S SUPERSEDE ISSUED=20-apr-2007 ck=173 \
SN=TK:5096-0:871303:843082 START=05-mar-2007 \
AUTH={ snpslmd=( LK=4D45D90FAD6B) \
avantd=( LK=A6F2F854633C) }

INCREMENT certify snpslmd 2012.12 30-dec-2014 1 \
VENDOR_STRING=altera,xilinx \
ISSUED=18-jun-2009 ck=222 SN=TK:255682:464396 \
START=18-jun-2009 AUTH={ synplctyd=( SIGN="0205 8DC3 BD05 CCBD \
28A7 6940 D88A 21EA 09EE 95EF 4B03 E74F EAAA 4165 BCD6 DF4E \
0CD1 DDE0 C9CC 7D50 56D0") }

INCREMENT spice snpslmd 2012.12 30-dec-2014 1 \
VENDOR_STRING=^1+S SUPERSEDE ISSUED=20-apr-2007 ck=184 \
SN=TK:5096-0:871303:843082 START=05-mar-2007 \
AUTH={ snpslmd=( LK=71C3F3F78009) \
sandwork=( SIGN=E3C45AB6E84E) }
```

Using the Key Name in Ascending Order and the Start Date in Descending Order

The following example uses the input file in.lic:

```
INCREMENT APSolar snpslmd 2012.12 30-dec-2014 1 \
   VENDOR_STRING=^1+S SUPERSEDE ISSUED=20-apr-2007 ck=173 \
   SN=TK:5096-0:871303:843082 START=05-mar-2009 \
   AUTH={ snpslmd=( LK=4D45D90FAD6B) \
   avantd=( LK=A6F2F854633C) }

INCREMENT certify snpslmd 2012.12 30-dec-2014 1 \
   VENDOR_STRING=altera,xilinx \
   ISSUED=18-jun-2009 ck=222 SN=TK:255682:464396 \
   START=18-jun-2009 AUTH={ synplctyd=( SIGN="0205 8DC3 BD05 CCBD \
   28A7 6940 D88A 21EA 09EE 95EF 4B03 E74F EAAA 4165 BCD6 DF4E \
   0CD1 DDE0 C9CC 7D50 56D0") }

INCREMENT APSolar snpslmd 2012.12 30-dec-2014 1 \
   VENDOR_STRING=^1+S SUPERSEDE ISSUED=20-apr-2007 ck=173 \
   SN=TK:5096-0:871303:843082 START=05-mar-2010 \
   AUTH={ snpslmd=( LK=4D45D90FAD6B) \
   avantd=( LK=A6F2F854633C) }
```

The example sorts the in.lic license key file by using the key name in ascending order and the start date in descending order.

```
% sclsort  -i in.lic  -p  feat:a –s startdate:d
```

This example prints the following sorted license key file to the file name out.lic.

```
INCREMENT APSolar snpslmd 2012.12 30-dec-2014 1 \
   VENDOR_STRING=^1+S SUPERSEDE ISSUED=20-apr-2007 ck=173 \
   SN=TK:5096-0:871303:843082 START=05-mar-2010 \
   AUTH={ snpslmd=( LK=4D45D90FAD6B) \
   avantd=( LK=A6F2F854633C) }
INCREMENT APSolar snpslmd 2012.12 30-dec-2014 1 \
   VENDOR_STRING=^1+S SUPERSEDE ISSUED=20-apr-2007 ck=173 \
   SN=TK:5096-0:871303:843082 START=05-mar-2009 \
   AUTH={ snpslmd=( LK=4D45D90FAD6B) \
   avantd=( LK=A6F2F854633C) }
INCREMENT certify snpslmd 2012.12 30-dec-2014 1 \
   VENDOR_STRING=altera,xilinx \
   ISSUED=18-jun-2009 ck=222 SN=TK:255682:464396 \
   START=18-jun-2009 AUTH={ synplctyd=( SIGN="0205 8DC3 BD05 CCBD \
   28A7 6940 D88A 21EA 09EE 95EF 4B03 E74F EAAA 4165 BCD6 DF4E \
   0CD1 DDE0 C9CC 7D50 56D0") }
```

D

Custom Server Debug Log

This appendix describes the Custom Server Debug Log feature and includes the following sections:

- Usage
- Custom Server Debug Logging Configuration File
- File Rotation Policy

snpslmd debug logging enables printing and logging extra debug information for check in, check out, and queuing requests. This functionality allows administrators and users to customize the format and order of debug logs, which can then be processed by custom scripts or programs.

Usage

You can enable or disable custom server debug logging with the sclsh tool, which is shipped with each SCL release.

```
sclsh> server_debug [on | off][custom file_path]
```

This section includes:

- Enabling Custom Server Debug Logging
- Disabling Custom Server Debug Logging

Enabling Custom Server Debug Logging

You can run the following sclsh commands to enable custom server debug logging.

```
% sclsh
sclsh> init
sclsh> server_debug on custom /path/server_debug_conf_file
```

The location of the server debug configuration file can be an absolute path or a relative path (relative to the directory from which the SCL server is started).

You must start custom server debug logging manually since it is not enabled by default. Custom server logging can be enabled or disabled using sclsh on the same machine (and by the same user who started the SCL Server) where the SCL Server is running.

You can stop custom server debug logging and start it again by providing any configuration file using the sclsh tool.

Disabling Custom Server Debug Logging

You can run the following sclsh commands to disable custom server debug logging.

```
% sclsh
sclsh> init
sclsh> server_debug off
```

Custom Server Debug Logging Configuration File

The custom server debug logging configuration file is an XML file. The following is an example of an XML configuration file:

Example D-1 XML Configuration File Example

```
<?xml version="1.0"?>
<CUSTOMLOG version="1.0">
        <LogFileProperty>
                <fileName>snpslmd.log</fileName>
                <delimiter>\t</delimiter>
                <rotate_policy>size</rotate_policy>
                <maxLogSize>100</maxLogSize>
                <prefix>(snpslmd)</prefix>
        </LogFileProperty>
        <Property value="status">
                <prefix>Command</prefix>
                <prolog>(</prolog>
                <epilog>)</epilog>
        </Property>
        <Property value="feature">
                <prefix>Feature Name</prefix>
                <prolog>(</prolog>
                <epilog>)</epilog>
        </Property>
</CUSTOMLOG>
```

Example D-1 includes the LogFileProperty Tag and the Property Tag. The following sections describe the LogFileProperty Tag, the Property Tag, and an XML Configuration File.

LogFileProperty Tag

The LogFileProperty Tag includes the following attributes shown in Table D-1.

Table D-1 LogFileProperty Tag Attributes

Attribute	Description
`fileName`	Name of the log file where customized debug logging information is written.
`delimiter`	The delimiter between each Property value in the Log file.
`rotate_policy`	The Log file rotation policy. Currently two types are supported: date or size.

Table D-1 LogFileProperty Tag Attributes (Continued)

Attribute	Description
`maxLogSize`	Log file rotation size in megabytes (MB). Only applicable if the rotation_policy is using the size type.
`prefix`	Each debug log message is prefixed with this string.

Property Tag

The configuration file can have multiple Property Tags identifying various fields that are required for Custom Debug Logging. The Property Tag consists of the following attributes shown in Table D-2.

Table D-2 Property Tag Attributes

Attributes	Description
`prefix`	User-defined alias of the property value.
`prolog`	User-defined string which prefix the property value.
`epilog`	User-defined string which gets postfixed for the property value.

Table D-3 lists the supported Property Tag fields. You can also refer to the configuration file example provided with the SCL Release.

Table D-3 List of Supported Property Tag Fields

SR No	Fields	Comment
1	`status`	(OUT, IN, DENIED, QUEUED) messages
2	`Date`	Date on server when licensing activity occurred
3	`Time`	Time on server when licensing activity occurred.
4	`feature`	Name of the Feature
5	`User`	Name of User on the client side
6	`Host`	Name of the Client Host
7	`numofkey`	Key quantity

Table D-3 List of Supported Property Tag Fields (Continued)

SR No	Fields	Comment
8	`onSupportVersion`	Checkout Version based on the OnSupport Flag
9	`display`	Client Display terminal
10	`flexlmVersion`	Client FLEXlm Version
11	`flexlmRevision`	Client FLEXlm revision
12	`clientPlatform`	Client OS Platform
13	`vendorDefinition`	Checkout Data associated with the checkout request
14	`pidofclient`	Client Process ID
15	`userProject`	Client Project (if LM_PROJECT is set)

XML Configuration File Example

The following is the custom logging output based on the default XML configuration file provided with the release.

```
(snpslmd)       (Command=OUT)   (Feature Name=F1)       (Time=0:36:59)
(User Name=user)     (Host Name=machine)    (Number of keys=4)
(On Support version=1.0)   (User display=/dev/pts/7)  (Flexlm Version=11)
(Flexlm Revision=9)  (Client Platform=i86_lsb) (Vendor definition=VD_XXX)
(PID of the client=15635) (User Project =LM_PROJ_XXX ) (Date =11/10/2013)

(snpslmd)       (Command=IN)    (Feature Name=F1)       (Time=0:37:5)
(User Name=user)     (Host Name=machine)    (Number of keys=4)
(On Support version= )  (User display=/dev/pts/7)     (Flexlm Version=11)
(Flexlm Revision=9)   (Client Platform=i86_lsb)(Vendor definition= )
(PID of the client=15635) (User Project =LM_PROJ_XXX ) (Date =11/10/2013)
```

Here is the behavior for the XML configuration file shown in Example D-1.

- The custom server debug log message is written in a debug log file named snpslmd.log.
- The output (log file entry) for each check in, check out, and queuing operation is:

  ```
  (snpslmd)       (Command=OUT)   (Feature Name=F1)
  ```

  - `(snpslmd)` is the prefix from the LogFileProperty tag
  - `(Command=OUT)` where,
    - (is the prolog

    - `Command` is the prefix
    - `OUT` is the status value for the command checkout
    - `)` is the epilog
- The order of the Property Tags printed in the log file depends on the sequence of entries in the XML configuration file. You can update the sequence of the Property Tags in the XML configuration file to change the printing order in the log file.

File Rotation Policy

The file rotation policy can be set to either size or date.

If the rotation policy is set to size, the file rotates when the size exceeds the size, specified in the XML configuration file (in megabytes). The rotated file is renamed as filename_1.log, filename_2.log, and so on.

If the rotation policy is set to date, the file rotates every calendar date. In this case, the rotated file name includes the date suffix in YYYY_MM_DD format. For example, filename_2016.06.07.log.

E

SCL License Server in a Cloud Environment

This appendix describes how to configure the SCL license server in a cloud environment.

- Using the SCL License Server
- Running FlexNet Licensing Service on Linux and Windows Systems

Using the SCL License Server

You can run the SCL license server in a cloud environment (for example, Amazon EC2, Microsoft Azure, and so on) for linux64 and Windows platforms.

To get a license file for the virtual machine, provide Synopsys with the VM_UUID host ID.

You can obtain the VM_UUID host ID with the lmhostid utility, which is shipped with the SCL release. Retrieve the host ID of the virtual machine as shown in Example E-1:

Example E-1 Retriving hostid With the lmhostid Utility

```
% lmutil lmhostid -ptype VM -uuid

lmutil - Copyright (c) 1989-2017 Flexera Software LLC. All Rights
Reserved.

The FlexNet host ID of this machine is
"VM_UUID=DF440538-8EB7-11DC-BBDA-FE7FE89E000F"
```

After you obtain a license file from Synopsys, verify that the host ID specified on the server line matches the VM_UUID host ID. For Example E-1, the license file would be formatted as follows:

```
SERVER hostname1 VM_UUID=DF440538-8EB7-11DC-BBDA-FE7FE89E000F 27020
VENDOR snpslmd /path/to/snpslmd
USE_SERVER
INCREMENT SSS...
INCREMENT F1...
```

Running FlexNet Licensing Service on Linux and Windows Systems

Before starting the SCL license server, you must first run the FlexNet licensing service on the SCL server host machine. The license server is unable to serve the keys if the FlexNet licensing service is not running.

Running the FlexNet Licensing Service on Linux

You need the following files to run the FlexNet licensing service:

- Install_fnp.sh: The script file starts the FlexNet licensing service to detect the VM host ID.
- FNPLicensingSERVICE/FNPLicensingSERVICE: The binary file is used by the install_fnp.sh script to start the Flexnet Licensing service.

All the files are available under the SCL_ROOT/linux64/bin directory. Use the following commands to start the FlexNet licensing service:

```
% cd /SCL_ROOT/linux64/bin
% install_fnp.sh -cert
```

If you get an error while installing the FlexNet licensing service, check whether the following FUSE packages are installed correctly on the virtual host.

- fuse
- fuse-devel-static
- gvfs-fuse
- libfuse2
- fuse-devel

Running the FlexNet Licensing Service on Windows

You need the installanchorservice.exe file to run the FlexNet licensing service. All the files are available in the SCL_ROOT/win32/bin directory.

Note:
You must have administrator rights to install the FlexNet licensing service on Windows.

To install the FlexNet licensing service, run the following command at the Windows command prompt.

```
installanchorservice <vendor> all
```

where *<vendor>* is your vendor name.

The FlexNet licensing service issues the following confirmation message (after installing the file)

- If a version of the FlexNet Licensing service already exists

```
Successfully installed anchor service for publisher <vendor>, product
all
Success code 2002: The new service is the same as, or older than, the
installed service so no upgrade is required.
```

- On a clean Windows system

```
Successfully installed anchor service for publisher <vendor>, product
all
Success code 2003: There was no service on the system so the new one
was installed.
```

To uninstall the FlexNet licensing service from a system with only one user registered, run the following command at the Windows command prompt.

```
uninstallanchorservice <vendor> all
```

where *<vendor>* is your vendor name.

The FlexNet licensing service issues the following confirmation message (after uninstalling the file):

```
Successfully uninstalled anchor service for publisher demo, product fnp
Success code 2004: The installed service has been successfully removed
from the system.
```

Index