Siemens Digital Industries Software Licensing Manual for Siemens EDA Products

Software Version 4.2

November 2024

SIEMENS

About Siemens Digital Industries Software

Siemens Digital Industries Software is a global leader in the growing field of product lifecycle management (PLM), manufacturing operations management (MOM), and electronic design automation (EDA) software, hardware, and services. Siemens works with more than 100,000 customers, leading the digitalization of their planning and manufacturing processes. At Siemens Digital Industries Software, we blur the boundaries between industry domains by integrating the virtual and physical, hardware and software, design and manufacturing worlds. With the rapid pace of innovation, digitalization is no longer tomorrow’s idea. We take what the future promises tomorrow and make it real for our customers today. Where today meets tomorrow. Our culture encourages creativity, welcomes fresh thinking and focuses on growth, so our people, our business, and our customers can achieve their full potential.

Support Center: **https://support.sw.siemens.com**

Send Feedback on Documentation: **https://support.sw.siemens.com/doc_feedback_form**

Table of Contents

1. Siemens Digital Industries Software Licensing for Siemens EDA Products Overview

Siemens EDA software uses Mentor Standard Licensing (MSL) for the Siemens Advanced Licensing Technology (SALT) system to administer software licenses. SALT MSL is based on FlexNet™ licensing and uses the FlexNet license file format. Siemens EDA uses FlexNet Publisher certificate-based licensing only.

Note

Siemens EDA uses the term *license file* instead of *certificate*.

SALT MSL has two implementations:

- SALT MGLS (Linux[1]) — Mentor Graphics Licensing System
- SALT PCLS (Microsoft® Windows®) — PC Licensing System

Licenses are delivered to you as a result of booking an order or requesting an evaluation. The authorization codes, which are in the license file, enable Siemens EDA applications.

This manual provides instructions and information for the system or license administrator on how to use SALT MGLS and SALT PCLS to license Siemens EDA software. For help getting started with SALT and the Siemens License Server and transitioning between previous product releases that continue to use legacy MSL and newer product releases that now require SALT MSL, refer to the Siemens License Server product page on Support Center at **https://support.sw.siemens.com/en-US/product/1586485382**.

Note

For instructions and information about installing and managing the Siemens License Server on supported systems using the Siemens License Server Installer, refer to the *Siemens Digital Industries Software License Server Installation Instructions*.

The "Siemens Digital Industries Software Licensing for Siemens EDA Products Overview" chapter discusses the following:

[1] Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

FlexNet Licensing Components

SALT licensing provides a way for FlexNet-enabled applications to acquire licenses from a licensing vendor daemon available on the network. The following are the FlexNet licensing components.

- **FlexNet-Enabled Application Program** — The application program uses MSL for SALT and communicates with the license server.
- **License Manager Daemon (lmgrd)** — The lmgrd daemon handles initial contact with the client application programs and passes the connection to the appropriate vendor daemon. The lmgrd daemon also starts and restarts the vendor daemons.

  Note

  Node-locked uncounted (mobile compute) licenses do not require lmgrd or a vendor daemon.

- **Vendor Daemon (saltd)** — The Siemens vendor daemon, saltd, which is the Common Vendor Daemon supporting all Siemens licenses, keeps track of the licenses that are checked out. If the vendor daemon process terminates for any reason, all users lose their licenses but usually regain them automatically when lmgrd restarts the vendor daemon.
- **License File** — The license file is a text file where FlexNet stores licensing data. Siemens Digital Industries Software creates this license file, which contains information about the server and vendor daemon and at least one line of data, called the INCREMENT line, for each licensed product.

  Each INCREMENT line contains an encryption code that is based on data on that line, the host ID of the server(s), and other Siemens Digital Industries Software supplied data such as expiration date, count, and version.

  For details about the license file's contents, see **License File Format**.

For more information about FlexNet components, refer to Revenera's *FlexNet Publisher License Administration Guide*.

Note

Siemens EDA uses FlexNet certificate-based licensing only.

Related Topics

Types of Licenses Available

Mentor Standard Licensing Benefits

SALT MSL has many benefits, including a standard implementation of FlexNet licensing, a customized licensing environment, debugging capabilities, and tools to help with license configuration and management.

- **Standard implementation of FlexNet Licensing for Siemens EDA applications** — SALT MSL provides a consistent licensing implementation across all Siemens EDA products and product lines, facilitating the management of application environments.
- **Licensing environment customization** — SALT MSL enables system administrators to control connection timeouts, license file and licensing software locations, and access to specified licenses. For more information, refer to **Environment Variables for Licensing**.
- **Debugging capabilities** — SALT MSL writes license information regarding check-out and check-in, queue, denial, and server status to a log file on the license server and client environment and transaction information in the application environment. For more information, refer to **lmgrd** and **SALT_LOGGING_DIR**.
- **Tools to help with license configuration and management** — To test license availability, SALT MSL provides the salt_mgls_ok / mgls_ok utilities for Linux and the salt_pcls_ok / pcls_ok and salt_mgls_ok / mgls_ok utilities for Windows. The mgls_ok and pcls_ok utilities check out mgcld licenses, and the salt_mgls_ok and salt_pcls_ok utilities check out saltd licenses. SALT MSL also supports standard FlexNet administration graphical user interface and command-line utilities. For more information, refer to **Utilities for Licensing**.

Related Topics

mgls_ok
pcls_ok
salt_mgls_ok
salt_pcls_ok

Types of Licenses Available

Various types of licenses can authorize Siemens EDA applications, and each has specific characteristics.

Floating

- Anyone on the network can use the license. The licenses are tied to a server host ID provided by a system host ID, network interface, or hardware key.
- A license server is required. The license server tracks how many licenses are available and how many are currently in use. You can use the options file to configure license servers to allow access only to certain groups or individuals.
- Licenses must be served from the computer whose host ID matches the host ID listed on the SERVER line of the license file.
- The license file contains both a SERVER and DAEMON line, and the quantity on each INCREMENT line is one or more.

Node-Locked Uncounted (Mobile Compute)

- For Windows only.

- The license is locked to a particular piece of hardware, either an Ethernet address or hardware key, and the hardware must be installed and operational for the license to work.
- The license file does not contain SERVER or DAEMON lines, and the quantity on each INCREMENT line is “uncounted.”
- A license server is not required.

Node-Locked Counted

- The license is locked to a particular piece of hardware, either an Ethernet address, host ID, or hardware key. The devices provide a unique identifier for the license.
- The software runs only on the system that has the particular piece of hardware installed.
- A license server is required. The license server tracks how many licenses are available and how many are currently in use. You can use the options file to configure license servers to allow access only to certain groups or individuals.
- The license file contains both a SERVER and DAEMON line, and the quantity on each INCREMENT line is one or more.
- The license must be served from the computer whose host ID matches the host ID on the SERVER line of the license file.
- The difference between floating and node-locked counted licenses is that node-locked counted licenses have a host ID field (HOSTID=) on each INCREMENT line.

Related Topics

License File Format
Vendor Daemon Options File

License File Format

Siemens Digital Industries Software delivers licenses to you as a result of booking an order or requesting an evaluation. The authorization codes, which are in the license file, enable Siemens EDA applications. The license file adheres to a standard format determined by your type of license.

Figure 1-1: License File Example

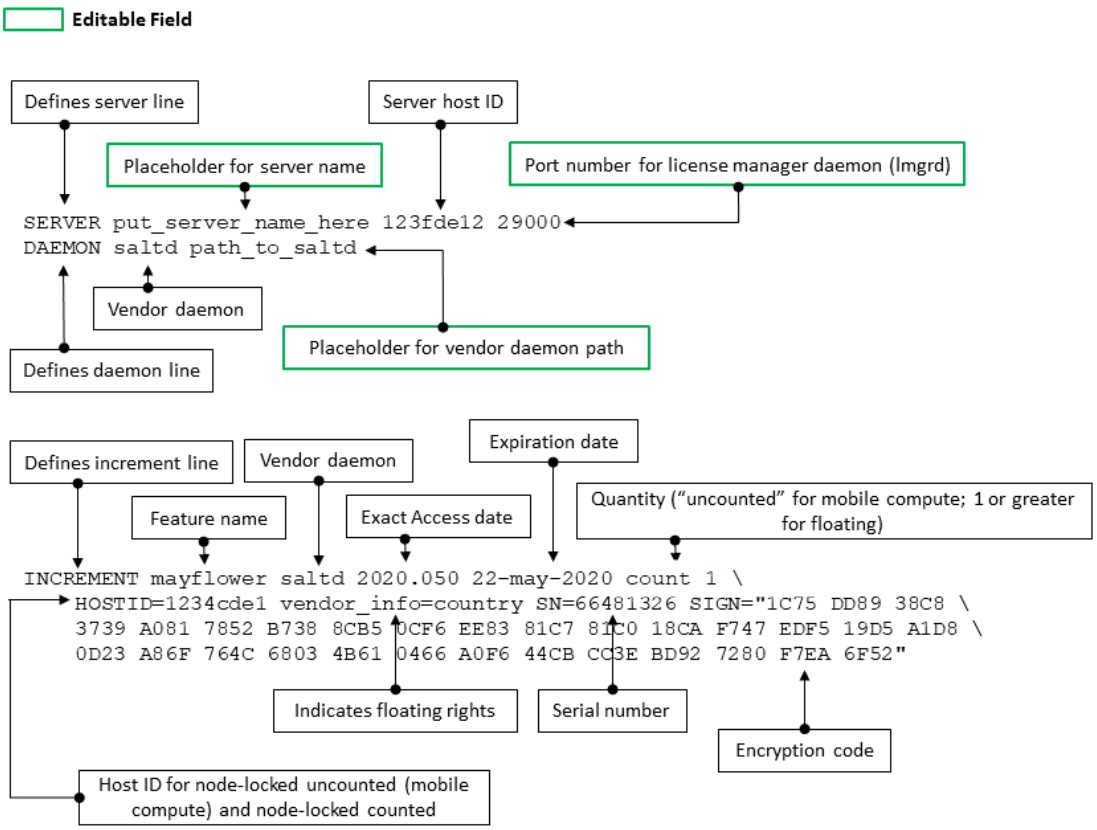


Figure 1-1 identifies the functional fields within a Siemens EDA license file. Optionally, you can add other fields. See Revenera's *FlexNet Publisher License Administration Guide* for more information. This example applies to all types of Siemens EDA licenses, so refer to the following descriptions of each field to determine which apply to your license environment.

SERVER Section

- **Defines server line** — (Floating and node-locked counted licenses only.) Tells the FlexNet license manager daemon (lmgrd) to expect information about the server.
- **Placeholder for server name** — (Floating and node-locked counted licenses only.) Shows a placeholder that you must replace with your server's hostname.

- **Server host ID** — (Floating and node-locked counted licenses only.) Shows the server host ID, which must match that of your server. The host ID can be an Ethernet (Linux and Windows) or a hardware key/dongle (Windows only). Do not change this field. To verify the host ID, use lmutil **lmhostid** or **LMTOOLS**. For additional methods, refer to **"Determining the Hostid for Licensing"** on the Siemens Digital Industries Software Licensing Community website.
- **Port number for license manager daemon (lmgrd)** — (Floating and node-locked counted licenses only.) Tells the license manager which TCP/IP port to use. You can change this to any available port.
- **Defines daemon line** — (Floating and node-locked counted licenses only.) Tells the license manager to expect information about a vendor daemon.
- **Vendor daemon** — (Floating and node-locked counted licenses only.) Tells the license manager that the information that follows applies to the Siemens vendor daemon.
- **Placeholder for vendor daemon path** — (Floating and node-locked counted licenses only.) Shows a placeholder that you must replace with the path to the vendor daemon, including the filename.

INCREMENT Section

- **Defines increment line** — Tells the license manager or license client to process the next fields as license feature information.
- **Feature name** — Defines the name of the feature, which the application checks out at run time.
- **Vendor daemon** — Designates this as a license feature that the Siemens vendor daemon or license client uses.
- **Exact Access date** — Shows the date relating to the version of the product you are currently running. For applications to run, your Siemens EDA products must have minimum license versions. Keeping the support contracts current for your products ensures that you will always be able to run the latest versions. For more information, refer to **"Exact Access Licensing"** on Support Center.
- **Expiration date** — Shows the date the license (authorization code) expires. This date corresponds to the end of your support contract term plus three months. For more information, refer to **"Authorization Code Duration Policy/Guideline"** on Support Center.
- **Quantity** — Defines the quantity of each license feature. For floating or node-locked counted licenses, the value is an integer greater than or equal to 1. For mobile compute licenses that are not counted by a license manager, the value is “uncounted.” This field helps you determine whether your licenses are floating or mobile compute. For more information about license types, refer to **"License Models, Servers, Types Policy Guideline"** on Support Center.
- **Host ID for node-locked uncounted (mobile compute) and node-locked counted** — Locks the feature to a specific host ID. The host ID can be an Ethernet (Linux and Windows) or a hardware key/dongle (Windows only). To verify the host ID, use lmutil **lmhostid** or **LMTOOLS**. For additional methods, refer to **"Determining the Hostid for Licensing"** on the Siemens Digital Industries Software Licensing Community website.
- **Indicates floating rights** — (Floating licenses only.) Reflects geographic usage rights defined by your license agreement.
- **Serial number** — Shows the unique ID that Siemens Digital Industries Software uses to look up license information.

- **Encryption code** — Shows the enhanced encryption for securing the license feature.

Determine Your Host ID for Licensing

You must provide your host IDs to Siemens Digital Industries Software to obtain the licenses for the products you have purchased.

You can determine your host ID by using lmutil **lmhostid** or **LMTOOLS**. For additional methods, refer to **"Determining the Hostid for Licensing"** on the Siemens Digital Industries Software Licensing Community website.

Additional Licensing Resources

Siemens Digital Industries Software provides a range of industry-leading support services that keep design teams productive and up-to-date with Siemens Digital Industries Software products.

A Siemens Digital Industries Software support contract includes the following:

- **Software Updates** — Get the latest releases and product enhancements to keep your environment current.
- **Siemens Digital Industries Software Support Center** — Access our online knowledge base, personalized to your Siemens Digital Industries Software products.
- **Support Forums** — Learn, share, and connect with other Siemens Digital Industries Software users.
- **Technical Support** — Collaborate with Siemens Digital Industries Software support engineers to solve complex design challenges.
- **Regular Communications** — Receive the latest knowledge base articles and announcements for your Siemens Digital Industries Software products.
- **Siemens Digital Industries Software Ideas** — Share ideas and vote for your favorites to shape future products.

More information is available here:

https://support.sw.siemens.com/

If your site is under a current support contract, but you do not have a Support Center login, register today:

https://support.sw.siemens.com/register

For answers to your account questions, refer to the **Account** page on Account Center.

For answers to frequently asked licensing questions and to collaborate with Siemens Digital Industries Software community members and technical experts, visit the Siemens Digital Industries Software Licensing Community website:

https://community.sw.siemens.com/s/topic/0TO4O000000QkHpWAK/licensing

2. Basic License Administration

SALT MSL provides the ability to install and configure your licensing software on Linux and Windows systems.

License Configuration and Management for Linux

Siemens Digital Industries Software provides the ability to configure and manage your licensing on Linux systems.

> **Note**
>
> Siemens EDA does not support hardware keys (dongles) on Linux systems.

Installing and Configuring a License Server on Linux

You must perform certain tasks to install licensing on a license server for the first time. For details, see the *Siemens Digital Industries Software License Server Installation Instructions*.

Configuring Licensing on a Client Workstation on Linux

After you start a license server, you can configure licensing on a client workstation.

Prerequisites

- You performed the procedures in the *Siemens Digital Industries Software License Server Installation Instructions*.

Procedure

1. Set the licensing environment variables to find the licensing software, and set the environment variables for the application to find the license file.

   You must set the SALT_LICENSE_SERVER variable to point to the license file location.

> **Note**
>
> A product may override the license server location that is specified by the SALT_LICENSE_SERVER environment variable.

2. Validate the server environment.

   Use lmstat to check that the server is running and a feature is available.

   ```
   $ lmutil lmstat -a -c {license_file | port@host}
   ```

   The lmstat command reports the server and daemon status and the product usage.

3. Use one of the MGLS utilities to verify that a client can check out a specific license feature.

   For use with the mgcld license file:

   ```
   $ mgls_ok feature_name
   ```

   For use with the saltd license file:

   ```
   $ salt_mgls_ok feature_name
   ```

Related Topics

SALT_LICENSE_SERVER
lmstat
mgls_ok
salt_mgls_ok

Adding a New License on Linux

You may need to add new licenses to your existing license server if, after you install your software and licenses, you decide to purchase additional Siemens EDA licenses for your existing software.

Prerequisites

- You installed the Siemens License Server by following the procedures in the *Siemens Digital Industries Software License Server Installation Instructions*.

Procedure

1. Set the licensing environment variables.

   You must set the SALT_LICENSE_SERVER variable so that the licensing software can find the location of the license file or server. For more information, refer to **Environment Variables for Licensing**.

Note

A product may override the license server location that is specified by the SALT_LICENSE_SERVER environment variable.

2. Edit the license file with the new licenses.

   Use a text editor to add the new INCREMENT lines to the existing license file.

3. Use lmreread to reread the license file if the server is running.

   ```
   $ lmutil lmreread -c {license_file | port@host}
   ```

4. Validate the server environment.

   Use lmstat to check that the server is up and that a feature is available.

   ```
   $ lmutil lmstat -a -c {license_file | port@host}
   ```

   This command reports the server and daemon status and product usage.

5. Use one of the MGLS utilities to verify that a client can check out a specific license feature.

   For use with the mgcld license file:

   ```
   $ mgls_ok feature_name
   ```

   For use with the saltd license file:

   ```
   $ salt_mgls_ok feature_name
   ```

Related Topics

License File Format
lmreread
lmstat
mgls_ok
salt_mgls_ok
Removing Expired Licenses From Your License File

Replacing a License File on Linux

You may need to replace a license file if you receive a renewed license.

Note

If you are managing a license file with term-based licenses, we recommend that you do not completely replace your license file. Instead, refer to Adding a New License on Linux.

Prerequisites

- You installed the Siemens License Server by following the procedures in the *Siemens Digital Industries Software License Server Installation Instructions*.

Procedure

1. Set the licensing environment variables.

   You must set the SALT_LICENSE_SERVER variable to the path of the license file. This enables the licensing software to find the location of the license file.

   Note

   A product may override the license server location that is specified by the SALT_LICENSE_SERVER environment variable.

2. Shut down the license manager.

   The license daemons write their last messages to the log file, close the file, and exit. All licenses that were checked out are rescinded. When the license manager restarts, the applications attempt to reacquire their licenses.

   Follow these steps:

   a. Use lmdown -c to shut down the license daemons and confirm the shutdown.

   ```
   $ lmutil lmdown -c {license_file | port@host}
   ```

   b. If you started the daemon with a server debug log file, check the log to ensure the daemons have exited.

   ```
   $ tail -f  logfile
   ```

   ```
   6/24 12:00(lmgrd)SHUTDOWN request from davem at node davem
   6/24 12:00(lmgrd)Shutting down saltd
   6/24 12:00(saltd)daemon shutdown requested-shutting down
   ```

3. Replace the license file.

   Rename the old license file and save the new license file in its place.

4. Copy the SERVER and DAEMON lines from the old license file and paste them in place of the SERVER and DAEMON lines in the new license file.

5. Start the license manager daemon on the license server.

   ```
   $ lmgrd -c license_file
   ```

   Where *license_file* is the file that you configured by following the procedures in the *Siemens Digital Industries Software License Server Installation Instructions*.

   You also can create a server debug log file.

```
$ lmgrd -c license_file -l logfile
```

Then you can use the Linux **tail -f** *logfile* command to monitor the status of the license server.

6. Verify that the server is valid and has started.

```
$ lmutil lmstat -a -c {license_file | port@host}
```

The lmstat command reports the server and daemon status and the product usage.

7. Verify that a client can check out a license.

Use one of the MGLS utilities to ensure that a client can check out a specific license feature.

For use with the mgcld license file:

```
$ mgls_ok feature_name
```

For use with the saltd license file:

```
$ salt_mgls_ok feature_name
```

Related Topics

Vendor Daemon Options File
SALT_LICENSE_SERVER
lmdown
lmgrd
lmstat
mgls_ok
salt_mgls_ok
Adding a New License on Linux

License Configuration and Management for Windows

Siemens Digital Industries Software provides the ability to configure and manage your licensing on Windows systems.

Installing and Configuring a License Server on Windows

You must perform certain tasks to install licensing on a license server for the first time. For details, see the *Siemens Digital Industries Software License Server Installation Instructions*.

Configuring Licensing on a Client Workstation on Windows

After you start a license server, you can configure licensing on a client workstation.

Prerequisites

- You performed the procedures in the *Siemens Digital Industries Software License Server Installation Instructions*.

Procedure

1. Set the licensing environment variables to find the licensing software, and set the environment variables for the application to find the license file.

   You must set the SALT_LICENSE_SERVER variable to point to the license file location. For information on how to set an environment variable, refer to "**How to set an environment variable on Windows**" on Support Center.

   Note

   A product may override the license server location that is specified by the SALT_LICENSE_SERVER environment variable.

2. Validate the server environment.

   Use lmstat to check that the server is running and a feature is available.

   ```
   C:> lmutil lmstat -a -c {license_file | port@host}
   ```

   The lmstat command reports the server and daemon status and the product usage.

3. Use either salt_pcls_ok (for the saltd license file) or pcls_ok (for the mgcld license file) to verify that a client can check out a specific license feature.

Related Topics

SALT_LICENSE_SERVER
lmstat
pcls_ok
salt_pcls_ok

Adding a New License on Windows

You may need to add new licenses to your existing license server if, after you install your software and licenses, you decide to purchase additional Siemens EDA licenses for your existing software.

Prerequisites

- You installed the Siemens License Server by following the procedures in the *Siemens Digital Industries Software License Server Installation Instructions*.

Procedure

1. Set the licensing environment variables.

   You must set the SALT_LICENSE_SERVER variable so that the licensing software can find the location of the license file or server. For more information, refer to **Environment Variables for Licensing**.

   > **Note**
   >
   > A product may override the license server location that is specified by the SALT_LICENSE_SERVER environment variable.

2. Edit the license file with the new licenses.

   Use a text editor to add the new INCREMENT lines to the existing license file.

3. Use lmreread to reread the license file if the server is running.

   ```
   C:> lmutil lmreread -c {license_file | port@host}
   ```

4. Validate the server environment.

   Use lmstat to check that the server is up and that a feature is available.

   ```
   C:> lmutil lmstat -a -c {license_file | port@host}
   ```

   This command reports the server and daemon status and product usage.

5. Use either salt_pcls_ok (for the saltd license file) or pcls_ok (for the mgcld license file) to verify that a client can check out a specific license feature.

Related Topics

License File Format
lmreread
lmstat
pcls_ok
salt_pcls_ok
Removing Expired Licenses From Your License File

3. Environment Variables for Licensing

You can control your licensing environment with environment variables, most of which are optional. The SALT_LICENSE_SERVER environment variable is required.

Note

Environment variables differ significantly between legacy Mentor Standard Licensing (MSL) and SALT MSL. For more information, refer to **Knowledge Base article MG612613** "Getting Started with Siemens Advanced Licensing Technology (SALT) and the Siemens License Server (SLS)" on Support Center.

The applications and utilities use the SALT_LICENSE_SERVER environment variable to determine the location of the license files or servers.

Note

To add, delete, or edit environment variables on Windows systems, refer to "**How to set an environment variable on Windows**" on Support Center.

Table 3-1: Environment Variable Summary

Environment Variable	Description
SALT_COLON_DELIM	On Linux, the SALT_COLON_DELIM environment variable specifies that colons (:) be used to separate multiple servers listed in the SALT_LICENSE_SERVER environment variable. By default, semicolons (;) separate licenses.
SALT_INCLUDE_LICENSES and SALT_EXCLUDE_LICENSES	The SALT_INCLUDE_LICENSES and SALT_EXCLUDE_LICENSES client-side environment variables let you control access to certain licenses.
SALT_LICENSE_SERVER	The SALT_LICENSE_SERVER client environment variable lists the license files or servers that Siemens EDA applications should use.
SALT_LICENSE_SOURCE	The SALT_LICENSE_SOURCE environment variable specifies the order in which licenses should be used in license check-out requests.
SALT_LOGGING_DIR	The SALT_LOGGING_DIR environment variable enables SALT licensing system debugging. Setting this variable creates a client debug log file.

SALT_COLON_DELIM

OS: Linux

On Linux, the SALT_COLON_DELIM environment variable specifies that colons (:) be used to separate multiple servers listed in the SALT_LICENSE_SERVER environment variable. By default, semicolons (;) separate licenses.

Examples

On Linux, the following example sets SALT_COLON_DELIM to use colons to separate multiple servers and then lists the license servers using SALT_LICENSE_SERVER:

```
SALT_COLON_DELIM=1
SALT_LICENSE_SERVER="license_file_pathname1:license_file_pathnameN:port@host
1:port@hostN"
```

SALT_INCLUDE_LICENSES and SALT_EXCLUDE_LICENSES

OS: Linux, Windows

The SALT_INCLUDE_LICENSES and SALT_EXCLUDE_LICENSES client-side environment variables let you control access to certain licenses.

These environment variables are similar to the FlexNet options file INCLUDE and EXCLUDE functionality, except that the restriction occurs on the client-side instead of the server-side.

Caution

Avoid mixing client-side variables with server-side restrictions (such as those in the FlexNet options file) for the same feature name. Doing so could result in failure to check out the required licenses.

While the SALT_INCLUDE_LICENSES and SALT_EXCLUDE_LICENSES environment variables are useful for controlling whether users can access certain licenses (for example: expadvpack_c), you should be careful when setting these variables. You could unknowingly restrict your machine from accessing any licenses. When possible, set these variables inside a script that invokes the application (for example: Xpedition® xPCB Layout) so that the restriction does not impact all Siemens EDA applications.

Example 1

The following example specifies to request only the expadvpack_c license.

Linux:

```
#!/bin/sh
SALT_INCLUDE_LICENSES=expadvpack_c
export SALT_INCLUDE_LICENSES
$SDD_HOME/common/linux/bin/ExpeditionPCB
```

Windows:

```
SET SALT_INCLUDE_LICENSES=expadvpack_c
START %SDD_HOME%\common\win32\bin\ExpeditionPCB.exe
```

Caution

When you set the SALT_INCLUDE_LICENSES environment variable, only the specified feature names can be requested. To prevent the value from affecting all Siemens EDA applications, set SALT_INCLUDE_LICENSES within a script rather than as a global environment variable.

Example 2

Linux:

The following example specifies to invoke xPCB Layout and have the tool ignore (exclude) the expadvpack_c license.

```
#!/bin/sh
SALT_EXCLUDE_LICENSES=expadvpack_c
export SALT_EXCLUDE_LICENSES
$SDD_HOME/common/linux/bin/ExpeditionPCB
```

Windows:

The following example has a mix of atomic xPCB Layout licenses (wgpcb, wgascentl2, wgascentlx, and so on), as well as the expadvpack_c composite. The example specifies to ignore the expadvpack_c composite and request the atomic licenses instead.

```
SET SALT_EXCLUDE_LICENSES=expadvpack_c
START %SDD_HOME%\common\win32\bin\ExpeditionPCB.exe
```

Caution

When you set the SALT_EXCLUDE_LICENSES environment variable, the specified feature names cannot be checked out. To prevent the value from affecting all Siemens EDA applications, set SALT_EXCLUDE_LICENSES within a script rather than as a global environment variable.

Example 3

Linux:

The following example specifies to never request an expadvpack_c or expadvtech_c license.

```
#!/bin/sh
SALT_EXCLUDE_LICENSES="expadvpack_c;expadvtech_c"
export SALT_EXCLUDE_LICENSES
$SDD_HOME/common/linux/bin/ExpeditionPCB
```

Windows:

The following example specifies to never request a padses_c, padsls_c or dxdesigner040_c license.

```
SET SALT_EXCLUDE_LICENSES="padses_c;padsls_c;dxdesigner040_c"
START %SDD_HOME%\common\win32\bin\ExpeditionPCB.exe
```

SALT_LICENSE_SERVER

OS: Linux, Windows

The SALT_LICENSE_SERVER client environment variable lists the license files or servers that Siemens EDA applications should use.

Caution

Make sure you specify the server information correctly; pointing to a nonexistent server or unresolvable hostname negatively impacts performance.

Note

Standard FlexNet Licensing commands such as lmutil, lmstat, and lmreread do not recognize the SALT_LICENSE_SERVER variable, so use the -c switch with those commands to explicitly point to the license location. The following is a Windows example. Note that when using FlexNet commands on Linux, you must use a colon instead of a semicolon as the separator.

```
lmutil lmstat -a -c "port@host;path_to_license_file"
```

SALT_LICENSE_SERVER Syntax

On both Linux and Windows, use a semicolon (;) to separate independent license sources. For example:

```
SALT_LICENSE_SERVER="license_file_pathname1;license_file_pathnameN;port@host
1;port@hostN"
```

Note

If you are using Linux and would like to use colons (:) instead of semicolons (;) to separate the license servers in SALT_LICENSE_SERVER, set the **SALT_COLON_DELIM** environment variable.

When you define multiple servers, the application uses the license servers in the order they are listed in the license file variable. For example, if you set your environment variable to the following:

```
SALT_LICENSE_SERVER="license_server1;license_server2"
```

the application requests a license from license_server1 and, if that license is in use or not available, the application requests the license from license_server2.

SALT_LICENSE_SERVER Length

The total length of the SALT_LICENSE_SERVER is dynamic and depends on the environment. In particular, the number of available file descriptors affects the number of license server or file entries. The basic licensing technology, FlexNet, limits each *port@host* entry to 1024 characters and a license path entry to 4096 characters. Your environment may have additional constraints.

Method to Reference a License Server

You can reference a license server by using the *port@host* syntax. This method enables you to contact a license server without needing to know the pathname to the license file that the license server is using.

The following example references an independent server.

```
SALT_LICENSE_SERVER=29000@host
```

The following example indicates the path to a group of redundant servers.

```
SALT_LICENSE_SERVER=29000@host1,29000@host2,29000@host3
```

Method to Reference a License File (Mobile Compute)

To reference a license file, use the pathname to the license file.

The following example indicates the pathname to the license file.

```
SALT_LICENSE_SERVER=C:\ProgramData\Siemens\License
Server\ActiveLicenses\saltd.lic
```

Method to Reference Multiple Server Configurations

To point to multiple server configurations, combine any of the syntax in a semicolon-separated list.

The following example points to multiple server configurations.

```
SALT_LICENSE_SERVER="29000@independent;29000@redundant1,29000@redundant2,290
00@redundant3"
```

SALT_LICENSE_SOURCE

OS: Linux, Windows

The SALT_LICENSE_SOURCE environment variable specifies the order in which licenses should be used in license check-out requests.

For example, you can use the SALT_LICENSE_SOURCE environment variable to ensure that a particular license is consumed before any other license.

Colons (on Linux) and semicolons (on Windows) separate the feature names in the list, and licensing searches through these names from left to right. In the environment variable, ANY indicates that all other licenses can be used.

Examples

Linux Example

The following Linux example specifies the xlayout201_c license and then the xlayout301_c license be consumed before any other license.

```
SALT_LICENSE_SOURCE=xlayout201_c:xlayout301_c
```

Windows Example

The following Windows example specifies the padses_c license be consumed before any other license.

```
SALT_LICENSE_SOURCE=padses_c:ANY
```

SALT_LOGGING_DIR

OS: Linux, Windows

The SALT_LOGGING_DIR environment variable enables SALT licensing system debugging. Setting this variable creates a client debug log file.

The client debug log file helps you troubleshoot basic licensing problems with client applications such as improperly installed licensing software, license files that cannot be located, or license files that are defined in multiple locations. The client debug log file does not contain any output from the server.

Note

Setting the SALT_LOGGING_DIR environment variable negatively impacts performance, so only turn on this variable for debugging purposes. Unset the SALT_LOGGING_DIR environment variable when troubleshooting is complete.

The client debug log file contains information about the following:

- Licensing environment variables
- Licensing runtime files (location and version)
- License file search paths
- Licensing errors
- Requested licenses, granted licenses, released (checked in) licenses, and validated licenses

Set the SALT_LOGGING_DIR environment variable to a directory where you want the client debug log files written. The directory must exist and be writable by the user invoking the client application, or no log file is generated.

The client debug log file is a text file with the process ID of the application embedded in the log filename (for example: *debug_log.7852.txt*). Each invocation of the application creates a new log file and does not overwrite nor append existing log files.

4. Troubleshooting Licensing Issues

If you experience licensing problems, first check that the license is available and that the environment variables are set correctly. Siemens Digital Industries Software provides additional help through various troubleshooting tools.

Caution

Changing your system's date and time may prevent Siemens EDA applications from checking out a license.

Check License Availability on Linux

Make sure you have the required license in a license file available for use. On Linux systems, running the salt_mgls_ok or mgls_ok utility attempts to check out and check back in a specified license and then reports success or failure.

Related Topics

mgls_ok
salt_mgls_ok

Check License Availability on Windows

Make sure you have the required license in a license file available for use. On Windows workstations, running the salt_pcls_ok or pcls_ok utility attempts to check out and check back in a specified license and reports any problems.

Related Topics

pcls_ok
salt_pcls_ok

Check the License Server Status

Run a status report to see server, daemon, and product usage.

Issue the following command:

```
lmutil lmstat -a -c {license_file | port@host}
```

The lmstat command performs the following actions:

- Checks license usage
- Reports the number of installed and available licenses for the specified product
- Lists all active licenses
- Lists all users of the specified feature
- Checks product availability

Caution

The frequency of lmstat requests and the volume of data each request returns can adversely affect the license server performance.

Related Topics

lmstat

Verify That the Environment Variables are Set Correctly

Make sure that the environment variables are set to the correct values. You must set at least the SALT_LICENSE_SERVER environment variable.

On Windows, you can use salt_pcls_ok or pcls_ok to check your licensing environment variables.

For details about the required and optional environment variables and their correct settings, refer to Environment Variables for Licensing.

Related Topics

mgls_ok
pcls_ok
salt_mgls_ok
salt_pcls_ok

Debug the Client Licensing Environment With the Client Debug Log File

Use the client debug log file to troubleshoot basic licensing problems with client applications. The license client creates this log file upon exiting if you have set the SALT_LOGGING_DIR environment variable to the path of a writable directory.

Related Topics

SALT_LOGGING_DIR

Common Client-Side Problems and Solutions

You may have problems starting an application that are related to Licensing. This section describes common issues and their solutions.

Table 4-2: Common Client-Side Problems Summary

Problem	Description
Version Not Supported	The license file/server does not support this version of the feature.
License Server is Not Defined	Licensing cannot locate the license server or license file.
Invalid Host	The host ID of this system does not match the host ID specified in the license file, the license for feature <feature_name> is not valid for this host ID, or an issue exists with the hardware key.
Vendor Daemon is Too Old	The version of the application being run requires a newer license server (lmgrd/saltd).
No License for the Feature	No license was found for the feature, no such feature exists, or the license server does not support this feature.
Cannot Connect to the License Server	The application cannot connect to the license server.
Application Slow to Launch	The application takes a long time to launch or open a project.
License Request Failure	The license request for a feature failed.
Cannot Acquire a License	Either all licenses for the feature are in use, not enough licenses are available for the feature, all licenses are in use and a request has been entered in all queues, or the license for the feature is not available yet.
Server/Client Connection Lost	Lost server/client connection for <feature_name>; attempt nnn in progress.
License Request Denial	License request denied because of SALT include/exclude lists.

Problem	Description
Licenses for the Feature are Unavailable	All <feature_name> licenses are in use, and queuing is disabled.
Package Information File Does Not Have the Requested License	The requested license ID is not in the package information file (mgc.pkginfo).
(Linux only) Unable to Start salt_mgls_asynch	The salt_mgls_asynch license process could not start.

Version Not Supported

The license file/server does not support this version of the feature.

Causes

The version of the license in the license file is older than the version the application requires.

Solution

Make sure the version (*YYYY.MM*0) on the INCREMENT line of the license file is greater than or equal to the version the application is requesting.

1. Set the SALT_LOGGING_DIR environment variable and examine the Transaction section of the client debug log file for information about the version of the license(s) the application requires.
2. Compare that information with the version of the licenses in the license file.

Check Account Center for your latest license file and, if the license file is not current, open a Support Case at https://support.sw.siemens.comsupport-case/open for a support renewal quote.

Related Topics

SALT_LOGGING_DIR
License File Format

License Server is Not Defined

Licensing cannot locate the license server or license file.

Causes

The SALT_LICENSE_SERVER environment variable is not set; SALT does not use the FlexNet default locations for the license file.

Solution

Set the SALT_LICENSE_SERVER environment variable to reference a license server containing the required licenses.

Related Topics

SALT_LICENSE_SERVER

Invalid Host

The host ID of this system does not match the host ID specified in the license file, the license for feature <feature_name> is not valid for this host ID, or an issue exists with the hardware key.

Solution

Table 4-3: Causes and Solutions

Cause	Solution
A mismatch exists between the host ID on the INCREMENT line of the license file and the host ID of the system running the application.	To verify whether the host IDs match, use the lmhostid command on Linux and the LMTOOLS utility on Windows. If the host ID does not match the host ID in the license file, open a Support Case at **https://support.sw.siemens.comsupport-case/open** to acquire a new license file.
The hardware key driver is not installed or the correct hardware key is not plugged in.	Install the hardware key driver and make sure the correct hardware key is plugged in and working.
The license you installed is sufficient for only one machine but not the current machine.	Run the software from the machine whose host ID matches the host ID specified at the server or in the license file. From **Account Center**, download and install the latest available licenses that are required to run the product software on your machine.

Related Topics

License File Format
lmhostid
LMTOOLS

Vendor Daemon is Too Old

The version of the application being run requires a newer license server (lmgrd/saltd).

Causes

One or more license servers specified in the SALT_LICENSE_SERVER environment variable are not up to date.

Solution

Make sure you are running the latest versions of the licensing software on your license server. For details, refer to Knowledge Base article **MG66951** on Support Center.

Related Topics

SALT_LICENSE_SERVER

No License for the Feature

No license was found for the feature, no such feature exists, or the license server does not support this feature.

```
License request for <feature_name> feature failed
```

Solution

Table 4-4: Causes and Solutions

Cause	Solution
The SALT_LICENSE_SERVER environment variable is not set or is not set correctly.	Before invoking the product, set the SALT_LICENSE_SERVER environment variable to reference your license server. For example: `SALT_LICENSE_SERVER=29000@lic_server` Contact your system administrator for settings information.

Cause	Solution
	Note A product may override the license server location that is specified by the SALT_LICENSE_SERVER environment variable.
The referenced server or license file does not contain the requested license feature.	Make sure the specified license server contains the requested license feature or that the correct license server or license file is specified in the SALT_LICENSE_SERVER environment variable. Set the SALT_LOGGING_DIR environment variable and examine the client debug log file for more information about licenses that the application is requesting but may not be able to find.
All licenses are in use and the product does not queue for licenses.	Wait for the license to become available before invoking the product.
The specified license file does not exist.	Verify that the license file exists and test the checkout of the feature name with salt_pcls_ok, pcls_ok, salt_mgls_ok, or mgls_ok.
The specified license file is corrupt.	Download and install your licenses from **Account Center**.

Related Topics

SALT_LICENSE_SERVER
SALT_LOGGING_DIR
mgls_ok
pcls_ok
salt_mgls_ok
salt_pcls_ok

Cannot Connect to the License Server

The application cannot connect to the license server.

```
FLEXnet Licensing error:-15,10. System Error: 10061 "WinSock: Connection
refused
```

Solution

Table 4-5: Causes and Solutions

Cause	Solution
The license server is down.	Verify that the server is running. Use lmutil lmstat on Linux or LMTOOLS on Windows.
The network is down or mis-configured.	Use standard network commands such as ping to check the network.
The system does not recognize the hostname in the license file.	Use the ping and nslookup commands to ensure that the hostname resolves to the correct IP address. If the network configuration requires a fully qualified domain name, include this on the SERVER line of the license file.
A firewall is blocking access to the license server.	Refer to **Firewall Considerations**.
The specified port number is incorrect.	Verify that the port number specified in the SALT_LICENSE_SERVER environment variable is the same as the port number defined on the SERVER line of the license file.
The lmgrd process is running on the license server, but the saltd process failed to start or has exited.	Make sure the vendor daemon path in the license file is the path to saltd and restart the license server.

Related Topics

lmstat
LMTOOLS
SALT_LICENSE_SERVER
License File Format

Application Slow to Launch

The application takes a long time to launch or open a project.

Solution

Table 4-6: Causes and Solutions

Cause	Solution
The SALT_LICENSE_SERVER environment variable includes license servers that are unreachable, not running, or nonexistent.	Verify that all servers in the list are running and reachable through the network. Remove any entries that do not resolve.
The network between the client and server has high latency, resulting in slow response time or a timeout while waiting for a license.	Address the high network latency issue between the client and the license server.

Related Topics

SALT_LICENSE_SERVER

License Request Failure

The license request for a feature failed.

Solution

Table 4-7: Causes and Solutions

Cause	Solution
The license server is down.	Restart the license server.
The SALT_LICENSE_SERVER environment variable is not set correctly.	Set the SALT_LICENSE_SERVER to reference a license server containing the required licenses. **Note** A product may override the license server location that is specified by the SALT_LICENSE_SERVER environment variable.
Some or all of the licenses your application requires have not been installed on your license server.	Download and install your licenses from **Account Center**.

Related Topics

SALT_LICENSE_SERVER

Cannot Acquire a License

Either all licenses for the feature are in use, not enough licenses are available for the feature, all licenses are in use and a request has been entered in all queues, or the license for the feature is not available yet.

Causes

Not enough licenses are installed at the referenced server(s) or license file(s).

All available licenses at the server(s) or license file(s) are checked out by other users. If your application allows your request to be queued, you may be placed in a queue for the required license(s), and you will be granted access when the license(s) become available.

Solution

- Download and install the latest available licenses for your products from **Account Center**.
- More license servers may be available than you are currently referencing; update SALT_LICENSE_SERVER to reference all available license servers.
- Purchase more licenses.

Related Topics

SALT_LICENSE_SERVER

Server/Client Connection Lost

Lost server/client connection for <feature_name>; attempt *nnn* in progress.

Causes

The application lost the connection to the referenced license server(s) and is attempting to reconnect.

Solution

If you receive the message “Reconnected to license server for feature <feature_name> after *nnn* retries,” the automatic reconnection attempts resolved the issue.

If the attempts are not successful, ensure that the license server is running and that you have a network connection to it.

License Request Denial

License request denied because of SALT include/exclude lists.

Causes

Either the SALT_INCLUDE_LICENSES or SALT_EXCLUDE_LICENSES environment variable is set in a way that prevents you from acquiring your licenses.

Solution

Unset SALT_INCLUDE_LICENSES or SALT_EXCLUDE_LICENSES and restart your application.

Related Topics

SALT_INCLUDE_LICENSES and SALT_EXCLUDE_LICENSES

Licenses for the Feature are Unavailable

All <feature_name> licenses are in use, and queuing is disabled.

Causes

The license(s) your application requires are currently in use by other users, and the application has specified that your license requests should not queue.

Solution

Open a Support Case at **https://support.sw.siemens.comsupport-case/open**.

Package Information File Does Not Have the Requested License

The requested license ID is not in the package information file (*mgc.pkginfo*).

Causes

The *mgc.pkginfo* file does not have the required information to run your software.

Solution

From Support Center, update the *mgc.pkginfo* file to the latest available version. For more information, refer to **KB article MG13664** "Licensing: Where can I find the latest version of the mgc.pkginfo file?"

(Linux only) Unable to Start salt_mgls_asynch

The salt_mgls_asynch license process could not start.

Solution

Table 4-8: Causes and Solutions

Cause	Solution
The salt_mgls_asynch binary does not have execute permissions.	Check the salt_mgls_asynch binary permissions to make sure it has execute permissions.
The salt_mgls_asynch binary has an older version than the application requires.	Update the salt_mgls_asynch binary to the same version as the *salt_msl.so* version.
Out of memory.	Open a Support Case at **https://support.sw.siemens.comsupport-case/open**.

Common Server-Side Problems and Solutions

You may have problems with the license server workstation. This section describes common issues and their solutions.

For additional help with server-side issues, refer to the “Troubleshooting” section in the *Siemens Digital Industries Software License Server Installation Instructions*.

Table 4-9: Common Server-Side Problems Summary

Problem	Description
License Server Does Not Support This Feature	UNSUPPORTED message displays in the server debug log file along with the message “License Server does not support this feature.”

License Server Does Not Support This Feature

UNSUPPORTED message displays in the server debug log file along with the message “License Server does not support this feature.”

Note

This message may be a hard-coded license check for the application. If the application does not show a problem, you can ignore the message.

Solution

Table 4-10: Causes and Solutions

Causes	Solution
A client application requests a license, incorrectly assuming the server has the license. The SALT_LICENSE_SERVER environment variable is set to a license file that is different from the license file the server is using.	Set the SALT_LICENSE_SERVER environment variable to *port@host*.
The license is not in the license file.	You will need to purchase the product and download the new license.

Related Topics

SALT_LICENSE_SERVER

Global Customer Support and Success

A maintenance contract with Siemens Digital Industries Software is a valuable investment in your organization's success. With a maintenance contract, you have 24/7 access to the comprehensive and personalized Support Center portal.

Support Center features an extensive knowledge base to quickly troubleshoot issues by product and version. You can also download the latest releases, access the most up-to-date documentation, and submit a support case through a streamlined process.

https://support.sw.siemens.com

If your site is under a current maintenance contract but you do not have a Support Center login, register here:

https://support.sw.siemens.com/register

A. Utilities for Licensing

Siemens EDA and FlexNet licensing utilities provide the ability to manage your licenses and license servers.

Syntax Conventions

The Syntax section of each utility's command description uses conventions that help clarify the command's usage.

Table A-11: Conventions for Command-Line Syntax

Convention	Example	Usage
Regular font	lmdown -all	Standard text indicates a literal command or argument that you enter.
Boldface	salt_mgls_ok ***feature_name***	Indicates a required argument.
Italic	-vendor *vendor_daemon*	Indicates a user-supplied argument.
[ ]	lmdown [-force]	Brackets enclose optional arguments. Do not enter the brackets.
{ }	lmreread -c {*license_file* | *port@host*}	Braces enclose arguments to show grouping. Do not enter the braces.
|	lmstat -c {*license_file* | *port@host*}	The vertical bar indicates an either/or choice between items. Do not include the bar in the command.

Utilities

Siemens EDA provides command-line and graphical user interface (GUI) utilities for Linux and Windows systems.

This section provides, in alphabetical order, reference information for Siemens EDA and *some* FlexNet command-line and GUI licensing utilities. The beginning of each utility description shows the applicable operating system.

For detailed information about the FlexNet Licensing utilities, refer to Revenera's *FlexNet Publisher License Administration Guide*.

Note

Mentor Standard Licensing currently does not support the lmadmin command. Use the commands listed in this appendix to manage your license server, and disregard the lmadmin documentation in the *FlexNet Publisher License Administration Guide*.

Table A-12: Utility Summary

Utility	Description
lmdown	(FlexNet) Shuts down selected license daemons on all systems in the license file list. This is useful when you need to make changes or updates to the licensing software.
lmgrd	(FlexNet) Starts a license server. lmgrd is the main license daemon program for FlexNet Licensing and handles requests from the end user's applications.
lmhostid	(FlexNet) Reports the host identification (host ID) number of a system.
lmreread	(FlexNet) Tells the license daemon to reread the license file and start any new vendor daemons that have been added. This command is useful for adding new licenses or for picking up simple changes to the options file such as defining a new LM_PROJECT.
lmstat	(FlexNet) Reports the status of all network licensing activities, including license files, daemons that are running, users of individual features, and users of features served by a specific vendor daemon.
LMTOOLS	(FlexNet) Starts, stops, and configures FlexNet license servers; retrieves system information, including host IDs; and obtains server status. While LMTOOLS performs other functions, these are the most commonly used.
mgls_admin	(SALT MGLS) Displays version information for the SALT MGLS environment.
mgls_ok	(SALT MGLS/SALT PCLS) For use with the mgcld license file only, mgls_ok checks out and checks in a specified feature and reports success or failure. You must set the SALT_LICENSE_SERVER environment variable before invocation.
pcls_ok	(SALT PCLS) For use with the mgcld license file only, pcls_ok checks out and checks in a specified feature and reports any problems. This application, which is useful for verifying the license file, reports the version of the licensing software and displays the environment settings for SALT_LICENSE_SERVER. You must set the SALT_LICENSE_SERVER environment variable before invocation.
pkgmap	(SALT PCLS) Displays Siemens EDA feature information for a specified product. You must set the SALT_LICENSE_SERVER environment variable before invocation.

Utility	Description
salt_mgls_ok	(SALT MGLS/SALT PCLS) For use with the saltd vendor daemon only, salt_mgls_ok checks out and checks in a specified feature and reports success or failure. You must set the SALT_LICENSE_SERVER environment variable before invocation.
salt_pcls_ok	(SALT PCLS) For use with the saltd vendor daemon only, salt_pcls_ok checks out and checks in a specified feature and reports any problems. This application, which is useful for verifying the license file, reports the version of the licensing software and displays the environment settings for SALT_LICENSE_SERVER. You must set the SALT_LICENSE_SERVER environment variable before invocation.

lmdown

OS: Linux, Windows

(FlexNet) Shuts down selected license daemons on all systems in the license file list. This is useful when you need to make changes or updates to the licensing software.

Caution

Using the lmdown command shuts down the specified license server and causes users to lose their licenses.

Syntax

lmutil lmdown -c {*license_file* | *port@host*} [-vendor *vendor_daemon*] [-q] [-all] [-force] [-help]

Arguments

- -c {*license_file* | *port@host*}

  Uses the specified license file, which is the full path to the license file, or uses the specified license server port (*port@host*). If you specify more than one component, you must separate each with a colon in Linux or a semicolon in Windows.

- -vendor [*vendor_daemon*]

  Shuts down only the specified vendor daemon. The lmgrd daemon continues running.

- -q

  Prevents issuing a prompt or printing a header.

- -all

  Shuts down all daemons if multiple servers are in use. The -q argument is implied with -all.

- -force

  If licenses are borrowed, restricts lmdown to run only from the system where the license server is running.

  Note

  Siemens EDA's implementation of FlexNet does not support license borrowing.

- -help

  Displays usage information and exits.

Description

The lmdown command sends a message to every license daemon specified on the command line requesting them to shut down. The license daemons write their last messages to the log file, close the file, and exit. All licenses that the daemons dispensed are rescinded. Therefore, the next time a client program attempts to verify the license, the license will not be valid.

Note

We recommend that you use lmdown or LMTOOLS to stop a license server. However, if lmgrd or saltd fails to exit, use the kill command on Linux systems or use the Task Manager to kill the processes on Windows systems.

You can use the lmdown command to shut down license servers configured for three-server redundancy. The servers shut down after a one-minute delay. To shut down only one of these license servers, you must shut down both the lmgrd and vendor daemon processes on that license server.

Examples

The following example uses the *license.txt* license file to shut down the license daemons.

lmutil lmdown -c license.txt

```
lmdown - Copyright (c) 1989-2013 by Flexera Software LLC.  All Rights
Reserved.

       Port@Host    Vendors
       1) 29000@mgcsrvr saltd

       Are you sure (y/n)? y
       1 FLEXnet License Server shut down
```

Related Topics

LMTOOLS

lmgrd

OS: Linux, Windows

(FlexNet) Starts a license server. lmgrd is the main license daemon program for FlexNet Licensing and handles requests from the end user's applications.

Syntax

lmgrd -c *license_file* [-l [+] *debug_log_path*] [-2 -p] [-local] [-x lmdown] [-x lmremove] [-z] [-v] [-help]

Arguments

- -c *license_file*

  Uses the specified license file, which is the full path to the license file. If you specify more than one license file, you must separate each with a colon in Linux or a semicolon in Windows.

- -l [+] *debug_log_path*

  Writes debugging information to the specified server debug log file. This option uses the letter l. The *debug_log_path* is the full path to the server debug log file. To append the logging entries, prepend the *debug_log_path* with a plus (+) sign.

- -2 -p

  Allows only a FlexNet Licensing administrator who is root by default to use the lmdown, lmread, or lmremove commands. If a privilege group called lmadmin exists, only members of that group can use the commands. The user root must be a member of the lmadmin group to use the commands. Using this option prevents non-privileged users from shutting down the license server with the lmdown command.

Note

We recommend that you do not run license servers as "root."

- -local

  Restricts lmdown to run from the same system as lmgrd.

- -x lmdown

  Disables the lmdown command. If lmdown is disabled, you must stop lmgrd by issuing the **kill** *pid* command on Linux or by pressing Ctrl-Alt-Del on Windows. You also must stop the lmgrd and vendor daemon processes. On Linux, do not use the -9 option with the kill command.

- -x lmremove

  Disables the lmremove command.

- -z

  (Windows only.) Runs lmgrd in the foreground. The machine displays separate windows for lmgrd and each vendor daemon. If you use the -l *debug_log_path* option, no windows are used.

- -v

  Prints lmgrd's version number and copyright and then exits.

- -help

  Displays usage information and exits.

Description

The lmgrd command searches for a license file that contains information about vendors and features and starts a license server. We recommend that you do not run license servers as root.

Examples

The following example uses the *license.txt* license file to start the license server and writes the debugging information to the *debug.log* file.

lmgrd -c license.txt -l debug.log

Related Topics

lmdown

lmhostid

OS: Linux, Windows

(FlexNet) Reports the host identification (host ID) number of a system.

Syntax

lmutil lmhostid [-n] [*type*] [-utf8]

Arguments

- -n

  Returns as a string only the host ID itself, which you provide to Siemens Digital Industries Software for license generation. The switch suppresses the header text.

- *type*

  Specifies the host ID type. The default is the host ID for the current platform.

  Valid host ID types are the following:

  - -ether — Ethernet address.
  - -flexid — Parallel or USB FLEXid hardware key identification. This applies only to platforms that support FLEXid hardware keys.

- -utf8

  Returns the host ID as a UTF-8 encoded string rather than as an ASCII string. Use this option if your host ID contains characters other than ASCII A through Z, a through z, or 0 through 9. To view a correct representation of the resulting host ID, use a text editor program that can display UTF-8 encoded strings.

Examples

The following requests the host identification number of the system and shows an example of the type of output you receive.

lmutil lmhostid

```
lmutil - Copyright (c) 1989-2020 Flexera. All Rights Reserved.
The FlexNet host ID of this machine is "0020afe6112a"
```

lmreread

OS: Linux, Windows

(FlexNet) Tells the license daemon to reread the license file and start any new vendor daemons that have been added. This command is useful for adding new licenses or for picking up simple changes to the options file such as defining a new LM_PROJECT.

Caution

If you replace the license file or make changes to the options file that involve restricting access to licenses, we recommend that you use the lmdown command and then restart the license server instead of using the lmreread command. Scheduling a restart helps mitigate problems that may result when you remove access for a user or host that already has the license checked out.

Syntax

lmutil lmreread -c {*license_file* | *port@host*} [-vendor *vendor*] [-all]

Arguments

- -c {*license_file* | *port@host*}

  Uses the specified license file, which is the full path to the license file, or uses the license server port (*port@host*). If you specify more than one component, you must separate each with a colon in Linux or a semicolon in Windows.

- -vendor *vendor*

  Specifies only one specific vendor daemon to reread the license file. Use the lmgrd command to restart the vendor daemon, if necessary.

- -all

  Instructs all instances of lmgrd to reread the license file if more than one lmgrd is specified.

Description

The lmreread command enables the system manager to tell the license daemon to reread the license file. If data in the license file has changed, the new data can be loaded into the license daemon without its needing to be shut down and restarted.

The license daemon always rereads the original loaded file. If you need to change the path to the license file, you must shut down and then restart the daemon with the new license file path.

You cannot use lmreread if the server name or port numbers have been changed in the license file. For these changes to take effect, you must shut down and restart the daemon.

If you specify a vendor daemon name, only that vendor's licenses and options file are reread.

Examples

The following example tells the license daemon to reread the *license.txt* license file and shows an example of the command results.

lmutil lmreread -c license.txt

```
lmutil lmreread - Copyright (c) 1989-2013 by Flexera Software LLC. All
Rights Reserved.
lmreread successful
```

Related Topics

lmgrd
lmdown

lmstat

OS: Linux, Windows

(FlexNet) Reports the status of all network licensing activities, including license files, daemons that are running, users of individual features, and users of features served by a specific vendor daemon.

Syntax

lmutil lmstat -c {*license_file* | *port@host*} [-a] [-f [*feature_name*]] [-i [*feature_name*]] [-S [*DAEMON*]] [-s [*server_name*]] [-t *timeout_value*] [-v] [-help]

Arguments

- -a

  Displays all information.

- -c {*license_file* | *port@host*}

  Uses the specified license file, which is the full path to the license file, or uses the license server port (*port@host*). If you specify more than one component, you must separate each with a colon in Linux or a semicolon in Windows.

- -f [*feature_name*]

  Displays all users of the specified feature. If you do not specify a feature, the command displays all usage information for all features.

- -i [*feature_name*]

  Displays information from the INCREMENT line for the specified feature. If you do not specify a feature, the command displays information for all features.

- -S [*DAEMON*]

  Lists all users of the specified daemon's features. If you do not specify a daemon, the command lists all users of all daemon's features.

- -s [*server_name*]

  Displays the status of all license files on the specified server. If you do not specify a server name, the command displays this status for all servers.

- -t *timeout_value*

  Sets the connection timeout to the specified value. This limits the amount of time lmstat spends attempting to connect to the specified server(s).

- -v

Displays the FLEXlm version, revision, and patch.

- -help

  Displays the usage information.

Description

The lmstat command provides information about the status of server systems, vendor daemons, or features and displays information received from the license server. The lmstat command does not provide information about the following:

- Unserved licenses
- Queued users
- Licenses shared due to duplicate grouping

To report on an uncounted license, the license must be added to a served license file and the application must be directed to use the license server for that license file (via @*host*, *port*@*host* or USE_SERVER).

Examples

The following example shows the results of using lmstat without any arguments; the system returns only the status and version information.

lmutil lmstat -c 29000@host1

```
lmstat - Copyright (c) 1989-2020 by Flexera Software LLC. All Rights
Reserved.
Flexible License Manager status on Fri 05/29/2020 10:39
License server status: 29000@host1
    License file(s) on host1: /opt/Siemens/LicenseServer/saltd.lic:
```

```
host1: license server UP (MASTER) v1.0.0
Vendor daemon status (on host1):
    saltd: UP v1.0.0
```

The following example shows the results of using lmstat to request a list of all users of the calibredrc feature.

lmutil lmstat -c 29000@host1 -f calibredrc

```
lmstat - Copyright (c) 1989-2020 by Flexera Software LLC. All Rights
Reserved.
Flexible License Manager status on Fri 05/29/2020 10:42
Users of calibredrc: (Total of 2 licenses issued; Total of 1 license in use)
```

```
   "calibredrc" v2014.08, vendor: saltd
   floating license
      macd myhost myhost (v2014.08) (host1/29000 102), start Fri 05/29 10:42
```

The following example shows the results of using lmstat to request the status of all network licensing activities.

Caution

Be aware that **lmstat -a** generates a large amount of server activity.

lmutil lmstat -c 29000@host1 -a

```
lmstat - Copyright (c) 1989-2020 by Flexera Software LLC. All Rights
Reserved.
Flexible License Manager status on Fri 05/29/2020 10:45
License server status: 29000@host1
   License file(s) on host1: /opt/Siemens/LicenseServer/saltd.lic:
host1: license server UP (MASTER) v1.0.0
Vendor daemon status (on host1):
   saltd: UP v1.0.0
Feature usage info:
Users of calibredrc: (Total of 2 licenses issued; Total of 1 license in use)
   "calibredrc" v2014.08, vendor: saltd
   floating license
      macd myhost myhost (v2014.08) (host1/29000 102), start Fri 05/29 10:42
Users of msimhdlmix: (Total of 2 licenses issued; Total of 0 licenses in
use)
```

LMTOOLS

OS: Windows

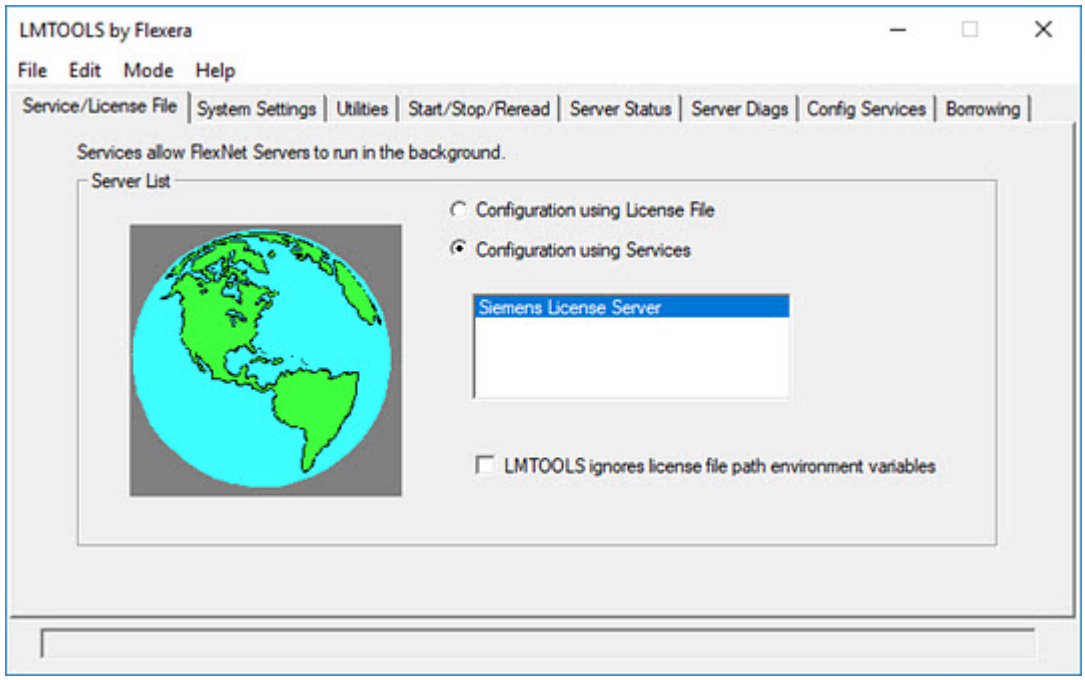
To access: From the **Start** menu, choose **Siemens License Server** > **Lmtools**

(FlexNet) Starts, stops, and configures FlexNet license servers; retrieves system information, including host IDs; and obtains server status. While LMTOOLS performs other functions, these are the most commonly used.

Description

LMTOOLS opens in the **Service/License File** tab. Select the “Configuration using Services” option.

Figure A-2: LMTOOLS — Service/License File Tab



This section describes the most commonly used tabs:

- **System Settings tab**
- **Start/Stop/Reread tab**
- **Config Services tab**

Note

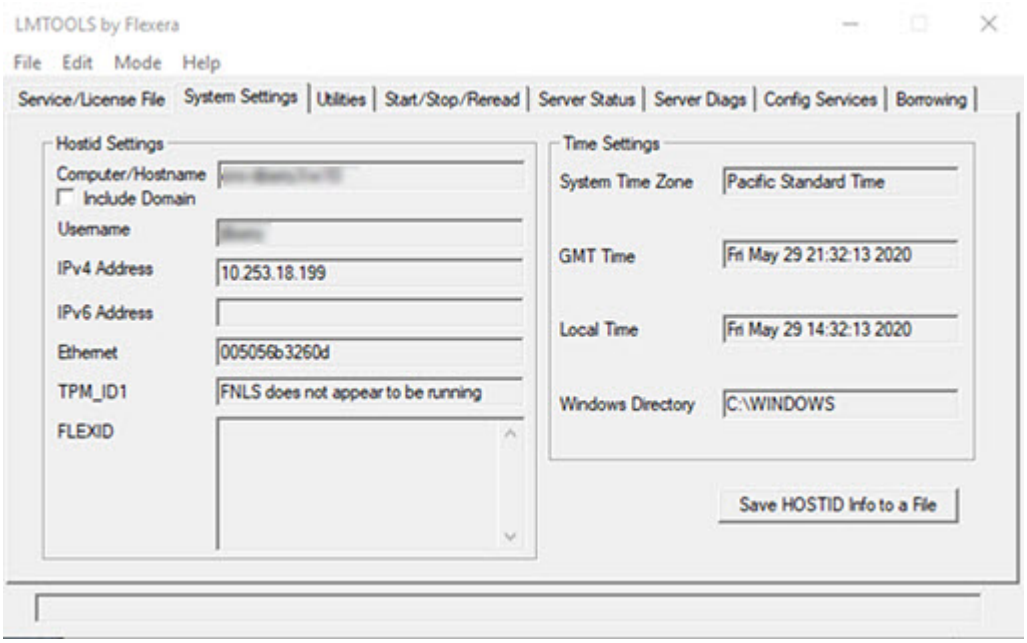
For a complete description of the LMTOOLS functionality, refer to the *FlexNet Publisher License Administration Guide*.

Objects

- **System Settings** tab

  Displays host ID and time settings for the current machine. This information is useful for obtaining host information, including the host id that licensing uses.

Figure A-3: LMTOOLS — System Settings Tab

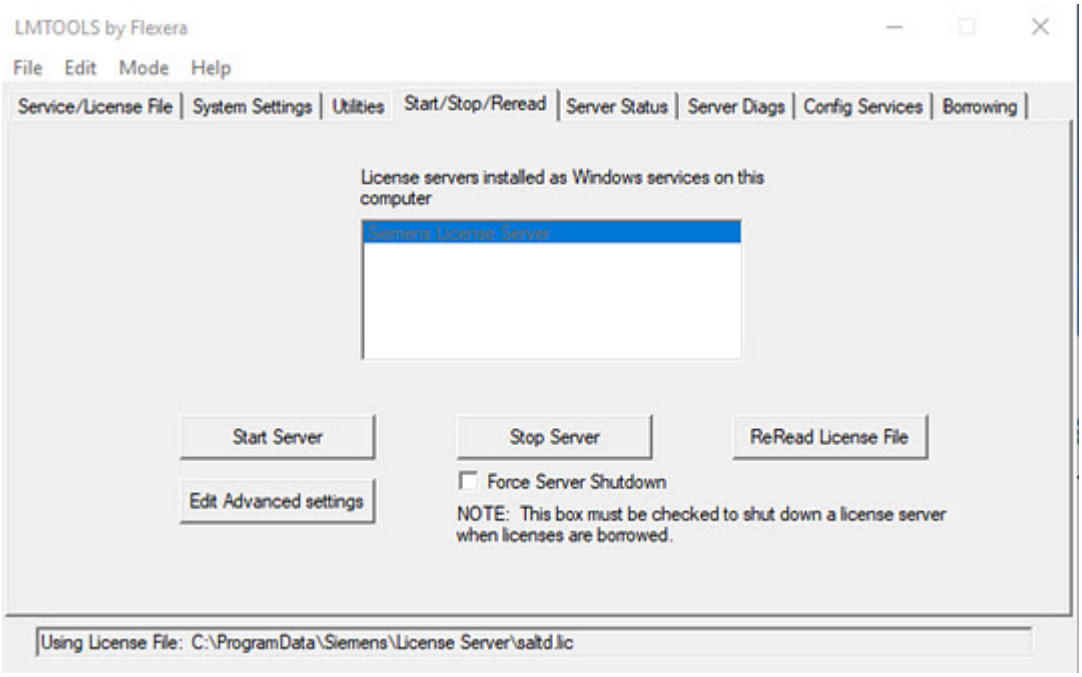


To save the information that displays in this tab, click the **Save HOSTID Info to a File** button and type a path and filename.

- **Start/Stop/Reread** tab

  Enables you to start or stop the server, or reread the license file.

Figure A-4: LMTOOLS — Start/Stop/Reread Tab



If you are adding new licenses or picking up simple changes to the options file such as defining a new LM_PROJECT, we recommend that you use the **ReRead License File** button. If you are replacing complete license files or making changes to the options file that involve restricting access to licenses, we recommend that you use the **Stop Server** and **Start Server** buttons. In the latter case, make sure you schedule the stop and start to mitigate any problems that could result if a user or host already has the license checked out.

- **Config Services** tab

  Configures lmgrd as a Windows service, which starts the license server automatically when you reboot your machine. You must have administrator privileges.

Figure A-5: LMTOOLS — Config ServicesTab

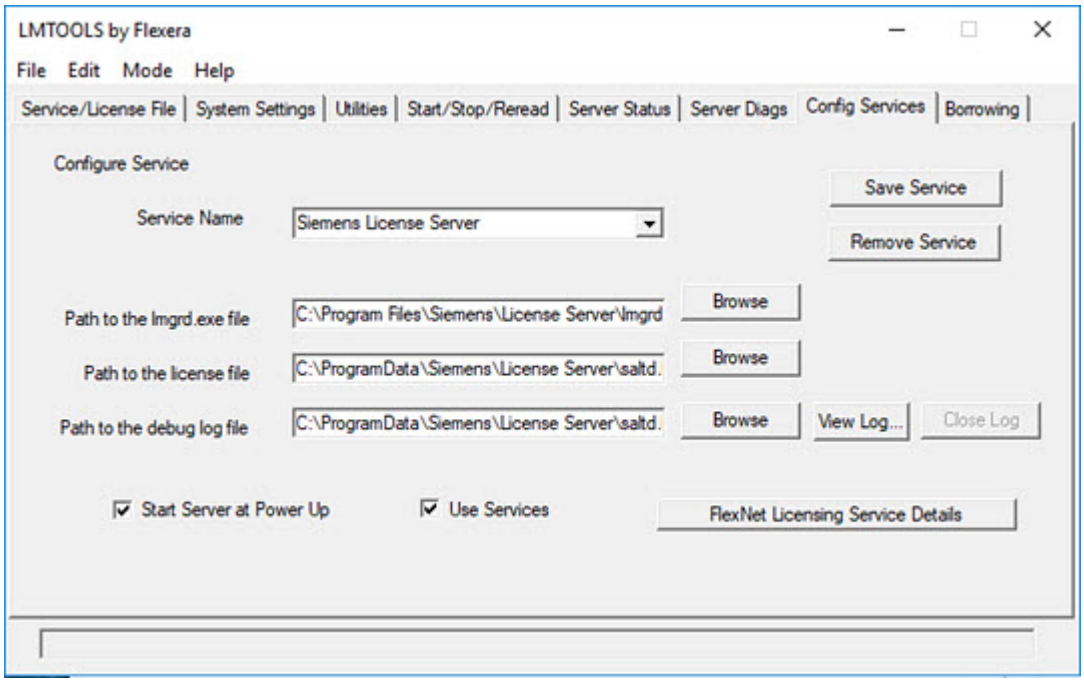


Table A-13: LMTOOLS — Config Services Tab Contents

Object	Description
Service Name field	Specifies the name of the service you want to define. The default name is FlexNet Publisher Service.
Path to the lmgrd.exe file field	Specifies the path and filename of the *lmgrd.exe* for this license server.
Path to the license file field	Specifies the path and filename of the license file for this license server.
Path to debug log file field	Specifies the path and filename of the server debug log file that this license server writes.
View Log button	Displays the server debug log file. On Windows, this is the only way you can view the log file.
Save Service button	Saves the new Windows service you just created.
Remove Service button	Removes the Windows service that is in the Service Name field. Click the down arrow in the Service Name field to select a different service.

Object	Description
Start Server at Power Up check box	Configures the license manager to start automatically as a Windows service at system reboot. This option is selectable only when you enable the Use Services option.
Use Services check box	Makes the license manager a Windows service.
FlexNet Licensing Service Details button	Not applicable to Siemens EDA licensing.

Related Topics

lmdown
lmhostid
lmreread
lmstat

mgls_admin

OS: Linux

(SALT MGLS) Displays version information for the SALT MGLS environment.

Syntax

mgls_admin [-v] [-pi *product_name*] [-help]

Arguments

- -v

  Displays version information for the SALT MGLS environment.

- -pi *product_name*

  Displays the Siemens EDA packaging information for the specified product. If *product_name* is an atomic feature name, mgls_admin lists the composites that contain the atomic. If *product_name* is a composite feature, mgls_admin lists the atomics in that composite.

- -help

  Provides additional information about the mgls_admin command. You can also use the shortened form of this switch, -h.

Examples

The following example shows version information for the SALT MGLS environment.

mgls_admin -v

```
mgls_admin: 1.0.0.0 Thu Nov 12 06:20:03 PST 2020
mgls_admin: package information
version v3.1_4.2i, written Wed Nov 4
17:01:10 2020
mgls_admin: mgls child version release:
v2020_1_patch1 version: v1.1_0
```

Related Topics

SALT_LICENSE_SERVER
pkgmap

mgls_ok

OS: Linux, Windows

(SALT MGLS/SALT PCLS) For use with the mgcld license file only, mgls_ok checks out and checks in a specified feature and reports success or failure. You must set the SALT_LICENSE_SERVER environment variable before invocation.

Syntax

mgls_ok [-v] [-silent] [-h] [-pd *product_date*] ***feature_name***

Arguments

- -v

  Displays the mgls_ok version.

- -silent

  Specifies to not display the results.

- -h

  Displays help text for mgls_ok.

- -pd *product_date*

  Requests a license that has the specified product date (*yyyy.mm*).

- ***feature_name***

  Specifies the name of the feature you want to check out and in. This is a required argument.

Examples

The following example successfully checks out the feature named msimhdlsim.

mgls_ok msimhdlsim

```
checking availability of "msimhdlsim"
license granted through "msimhdlsim"
```

Related Topics

SALT_LICENSE_SERVER

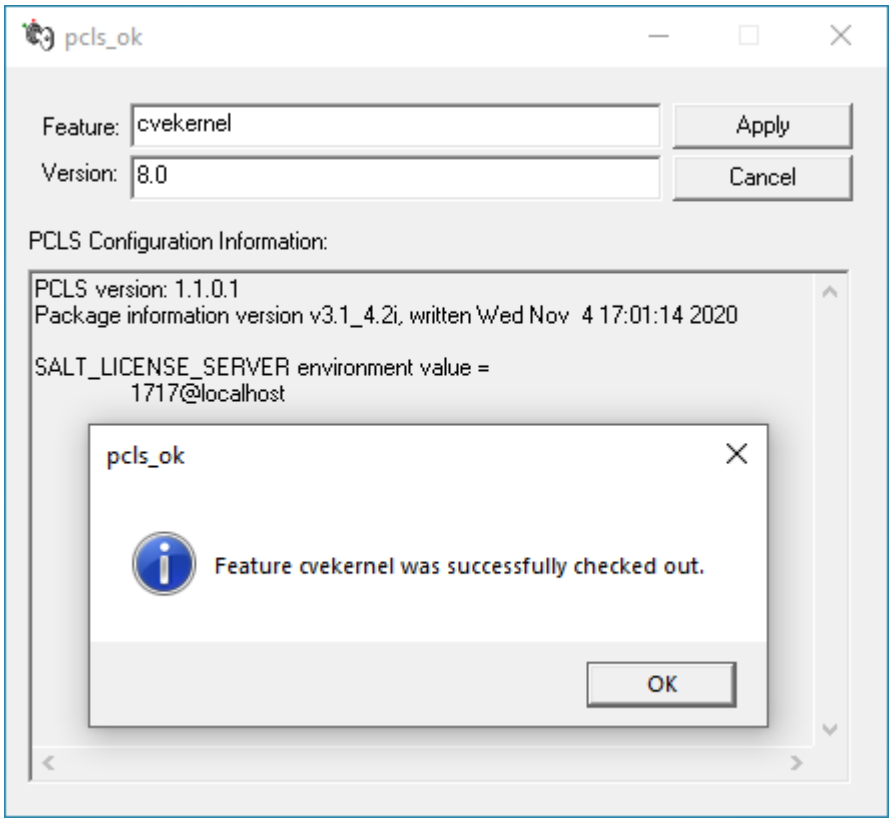
pcls_ok

OS: Windows

(SALT PCLS) For use with the mgcld license file only, pcls_ok checks out and checks in a specified feature and reports any problems. This application, which is useful for verifying the license file, reports the version of the licensing software and displays the environment settings for SALT_LICENSE_SERVER. You must set the SALT_LICENSE_SERVER environment variable before invocation.

Description

Figure A-6: pcls_ok Dialog Box



Fields

- Feature

  The name of the feature. This is the license name of the application you are trying to verify. In your license file, the feature name is to the right of the word INCREMENT.

- Version

  (Optional) The version number of the application you are trying to verify. For more information, see “Exact Access Date” in License File Format.

Usage Notes

The pcls_ok application attempts to check out and check back in the license you specify and generates a success or failure message. If you receive a success message, the licenses can be checked out successfully from that workstation. If the license checkout fails, a dialog opens that provides you with more information about the cause of the failure.

Related Topics

SALT_LICENSE_SERVER

pkgmap

OS: Windows

(SALT PCLS) Displays Siemens EDA feature information for a specified product. You must set the SALT_LICENSE_SERVER environment variable before invocation.

Syntax

pkgmap **-pi *feature*** [-p *path_to_mgc.pkginfo*]

Arguments

- **-pi *feature***

  Displays the Siemens EDA packaging information for the specified product. If *feature* is an atomic feature name, pkgmap lists the composites that contain the atomic. If *feature* is a composite feature, pkgmap lists the atomics in that composite.

- -p *path_to_mgc.pkginfo*

  Specifies the location of the *mgc.pkginfo* file. The default search path is as follows:

Sequence	Search Path
1	Path from the command line
2	Same location as the pkgmap executable
3	*%PATH%*

Examples

The following example shows composite information for an atomic feature.

pkgmap -pi variantmgrpads

```
Package information version v3.1_3.5i, written Sat Nov 03 17:01:12 2018
```

```
pkgmap: variantmgrpads
 variantpgrpads is contained in the following composites...
  padses_c
  padsds_c
  padsbr_c
  exppe_c
  exppeat_c
  padspro_c
  padsstdplus_c
```

```
  padsams_c
  variantmgrpro_c
  padspropcb_c
```

The following example shows atomic information for a composite feature.

pkgmap -pi padsams_c

```
Package information version v3.1_3.5i, written Sat Nov 03 17:01:12 2018
```

```
pkgmap: composite padsms_c
padsams_c {
  dxdatabook
  dxpdf
  visualibis
  hyplsw
  hyplswxtk
  netlistflow
  variantmgrpads
  viewdraw040
  dxdblocal
  cespads
  lmpads
  padsprjint2
} padsams_c composite  ENDS
```

The following example shows composite information for an atomic feature in a specified *mgc.pkginfo* file.

pkgmap -p mgc.pkginfo -pi pdq

```
Package information version v3.1_3.5i, written Sat Nov 03 17:01:12 2018
```

```
pkgmap: pdq
 pdq is contained in the following composites...
  ttraces_c
  xtraces_c
  xtraces3d_c
  sclin0003ln_c
  bsdall_c
```

Related Topics

SALT_LICENSE_SERVER
mgls_admin

salt_mgls_ok

OS: Linux, Windows

(SALT MGLS/SALT PCLS) For use with the saltd vendor daemon only, salt_mgls_ok checks out and checks in a specified feature and reports success or failure. You must set the SALT_LICENSE_SERVER environment variable before invocation.

Syntax

salt_mgls_ok [-v] [-silent] [-h] [-pd *product_date*] ***feature_name***

Arguments

- -v

  Displays the salt_mgls_ok version.

- -silent

  Specifies to not display the results.

- -h

  Displays help text for salt_mgls_ok.

- -pd *product_date*

  Requests a license that has the specified product date (*yyyy.mm*).

- ***feature_name***

  Specifies the name of the feature you want to check out and in. This is a required argument.

Examples

The following example successfully checks out the feature named msimhdlsim.

salt_mgls_ok msimhdlsim

```
checking availability of "msimhdlsim"
license granted through "msimhdlsim"
```

Related Topics

SALT_LICENSE_SERVER

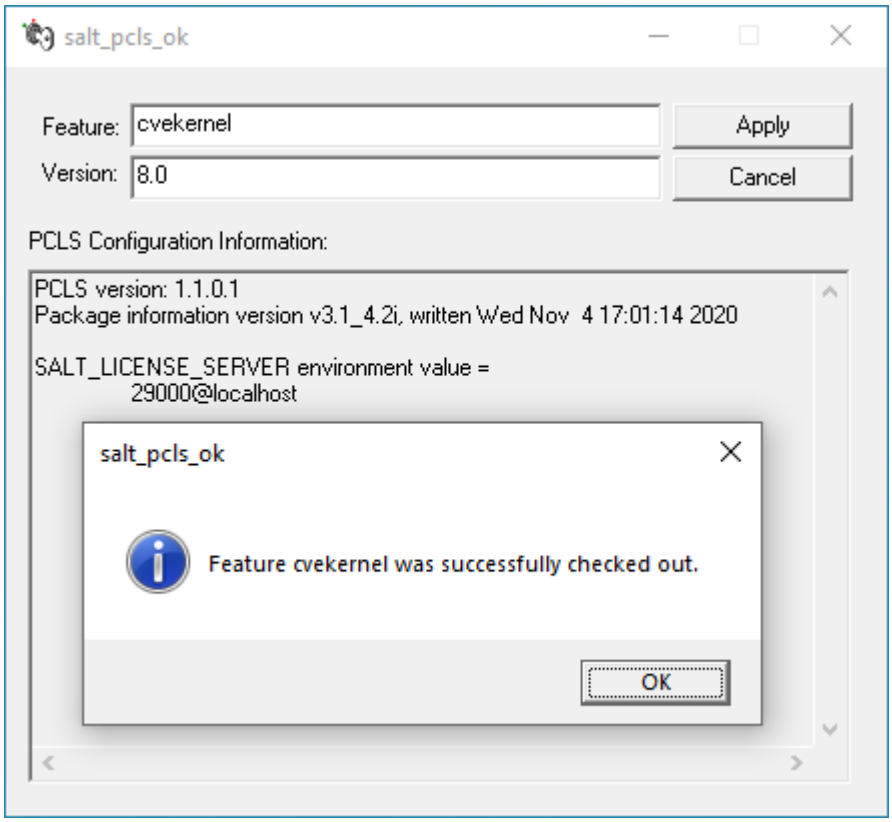
salt_pcls_ok

OS: Windows

(SALT PCLS) For use with the saltd vendor daemon only, salt_pcls_ok checks out and checks in a specified feature and reports any problems. This application, which is useful for verifying the license file, reports the version of the licensing software and displays the environment settings for SALT_LICENSE_SERVER. You must set the SALT_LICENSE_SERVER environment variable before invocation.

Description

Figure A-7: salt_pcls_ok Dialog Box



Fields

- Feature

  The name of the feature. This is the license name of the application you are trying to verify. In your license file, the feature name is to the right of the word INCREMENT.

- Version

  (Optional) The version number of the application you are trying to verify. For more information, see “Exact Access Date” in **License File Format**.

Usage Notes

The salt_pcls_ok application attempts to check out and check back in the license you specify and generates a success or failure message. If you receive a success message, the licenses can be checked out successfully from that workstation. If the license checkout fails, a dialog opens that provides you with more information about the cause of the failure.

Related Topics

SALT_LICENSE_SERVER

B. Advanced Topics

You may have need to restrict or customize license usage, determine the order of license checkout, make accommodations so that your company's firewall does not interfere with licensing communications, or manually remove expired licenses from your license file.

Vendor Daemon Options File

Use the vendor daemon options file to restrict or customize license usage.

You can perform activities such as the following:

- Exclude an individual, group, or IP address range from using one particular INCREMENT line.
- Reserve an INCREMENT for an individual.

Note

You can enable simple changes to the options file, such as defining a new LM_PROJECT, with the lmutil lmreread command. However, if you make changes to the options file that involve restricting access to licenses, such as using RESERVE, you should shut down the license server with the lmutil lmdown command and then restart the license server. Scheduling a restart helps mitigate problems that may result when you remove access for a user or host that already has the license checked out.

The vendor daemon options file is a text file usually located in the same directory as your license file. This file contains FlexNet Licensing-specific options.

If you want to use the vendor daemon options file, type the pathname as the fourth field on a DAEMON line in your license file. Otherwise, omit the fourth field.

For descriptions of the keywords in the vendor daemon options file and the requirements for their use as well as options file examples, refer to Revenera's *FlexNet Publisher License Administration Guide*.

Related Topics

License File Format
lmreread
lmdown

License Ordering in a License File

The order of licenses in a license file can affect the order of license checkout.

For example, if Feature2 displays before Feature1 in the license file and both licenses can satisfy the license request, Feature2 would usually be checked out first, if available.

If multiple licenses are for the same feature name and each has different license attributes, the licensing software automatically sorts the licenses according to the following criteria:

- Node-locked before floating
- Earlier versions before later versions

You can override this order by adding the *sort=* attribute. For more information, refer to Revenera's *FlexNet Publisher License Administration Guide*.

Related Topics

License File Format

Firewall Considerations

Firewalls can interfere with licensing communication.

On Linux systems, we recommend that you specify a static port number for the vendor daemon. You can do this by specifying PORT=*keyword* on the DAEMON line in your license file. For example:

```
DAEMON saltd saltd PORT=29001
```

or

```
DAEMON saltd /opt/mgls/bin/saltd PORT=29001
```

After you set the vendor daemon to run on a specific port, open that port on your firewall. For instructions on opening ports, refer to your firewall documentation.

On Windows systems, we recommend that you make exceptions for the licensing processes *lmgrd.exe* and *saltd.exe*, as well as any other vendor daemons that might be running. To make exceptions to processes, consult your Network Administrator.

Note

The Siemens License Server Installer can make this change if you do not want to do so manually. Refer to "Installing the License Server" in the *Siemens Digital Industries Software License Server Installation Instructions*.

Removing Expired Licenses From Your License File

If you replace your license file with a renewed license file, you may need to manually edit the file to remove expired or soon-to-expire licenses.

This avoids receiving warning messages about expired licenses. Also, some applications error out if they detect expired licenses.

Procedure

1. Open your license file in a text editor.
2. Either select and delete the expired or soon-to-expire INCREMENT lines or comment them out with the pound sign (#).

   For example:

```
SERVER server1 abcd1234abcd 29000
DAEMON saltd saltd
#INCREMENT padses_c saltd 2014.090 2-oct-2014 5 SN=50387478 \
#  SIGN="1FA7 FB2E 8DD6 B287 C11A 2E21 1A99 B3C5 9A71 61F0 7C26 FC88 \
#  79BC E952 09CC 0248 CB29 0DE0 C288 B47D BBF7 5CC6 6A23 2F9C 691B \
#  DEE0 3DE6 8830 595E D6CB"
INCREMENT padses_c saltd 2015.090 3-oct-2016 5 SN=50401278 \
  SIGN="1021 98B8 46A0 E391 CEB1 AF07 873B 30ED 4D02 3B98 D5B3 4817 253A \
  1B40 BE50 03B3 8DB7 3CA0 5779 8B4D 787C 7F9D 8D7B 8D54 46D7 6F1A 6416 \
  59EC 222B 8D99"
```

3. Save your license file.
4. If you are using a mobile compute license, restart your application. If you are running a license server, use the lmutil lmreread command.

Related Topics

License File Format
lmreread

Glossary

atomic license

A type of license that authorizes the use of a single function or a single application. A composite license contains atomic licenses.

authorization code

The license that enables the Siemens EDA software and related documentation for a certain period of time.
See also **INCREMENT line**

client

The host requesting a license from a server.

cluster servers

see **redundant servers**

composite license

A type of license that groups atomic licenses and usually ends in either an "_c" or "_s" suffix. A composite license typically authorizes a set of functionality within an application or across multiple applications.

counted (served) license

A license configuration that uses a license manager. Counted licenses can be floating, where they can be checked out from any host on the network, or they can be node-locked, where they are generated for use by a specific computer. The server can be either redundant or independent.

Exact Access Date

The date encoded in an INCREMENT line. Allows access to software updates for existing licensed software released prior to a support contract expiration date.

feature name

The name of the license that the application checks out at runtime. The feature name is a field on the INCREMENT line in a license file. Feature names are either composite or atomic licenses.

FlexNet Licensing

The license manager that Revenera provides. The SALT licensing software uses FlexNet Licensing as the technology for the licensing system.

floating license

A type of license that any workstation can check out.

hardware key/dongle

A device that plugs into a USB port and provides a unique host ID for the licensing system (Windows only).

host ID

A unique hardware-based number for each machine. The host ID ties licenses to a specified server or workstation.

INCREMENT line

The actual license that enables a product in the license file. The INCREMENT line includes fields that provide license feature information. *See also* **authorization code**

independent server

A type of server that contains a unique set of license data. If an independent server is off the network, all licenses associated with the server are unavailable.

license manager daemon

A generic term for one of two processes that runs on a license server and waits for requests from client applications. SALT licensing includes two daemons: lmgrd and saltd.

license file

A text file that contains the customer site identification, license server and daemon information, and one or more INCREMENT line, which is the actual license for the supported products.

license manager

see **license server**

license server

A workstation that runs the license daemons and provides licenses to client workstations.

lmgrd

The FlexNet license manager daemon.

mgcld

The license type for Siemens EDA products using the older mgcld encryption. The saltd vendor daemon with Common Vendor Daemon capability now manages these licenses.

mgc.pkginfo

The file that defines the mapping between atomic licenses and composite licenses. This mapping is what creates bundles of license features in the form of a composite license that can be checked out as a complete set of product functionality. For added detail, set the **SALT_LOGGING_DIR** environment variable.

mobile compute license

see **node-locked uncounted license**

node-locked license

A type of license that is tied to a particular system.

node-locked uncounted license

A type of license (Windows only) that is locked to a hardware key or Ethernet address of a workstation. This type of license, sometimes referred to as a mobile compute license, does not require a license server or network connection.

redundant servers

Types of servers that are configured in a group of three license servers that contain the same license data. Redundant servers, also called cluster or triad servers, enable access to all licenses on the servers as long as at least two of the three servers are operational.

Revenera

The software company, formerly Flexera Software, that produces FlexNet Licensing.

SALT MGLS

Siemens EDA licensing system for Siemens Advanced Licensing Technology. This software, which is for Linux, contains the lmgrd, saltd, and other utilities and programs to assist the system administrator with the license server setup.

SALT MSL

Mentor Standard Licensing (MSL) for Siemens Advanced Licensing Technology (SALT). SALT MSL has two implementations: SALT MGLS and SALT PCLS.

SALT PCLS

PC licensing system for Siemens Advanced Licensing Technology, which is essentially SALT MGLS on Windows.

saltd

The Siemens vendor daemon and a child of the lmgrd parent. The lmgrd and saltd daemons run the license server and handle requests from the end-user's applications.

server ID

The host ID of the license server machine.

triad servers

see **redundant servers**

vendor daemon

The daemon that dispenses licenses for the requested feature. The Siemens vendor daemon is saltd.