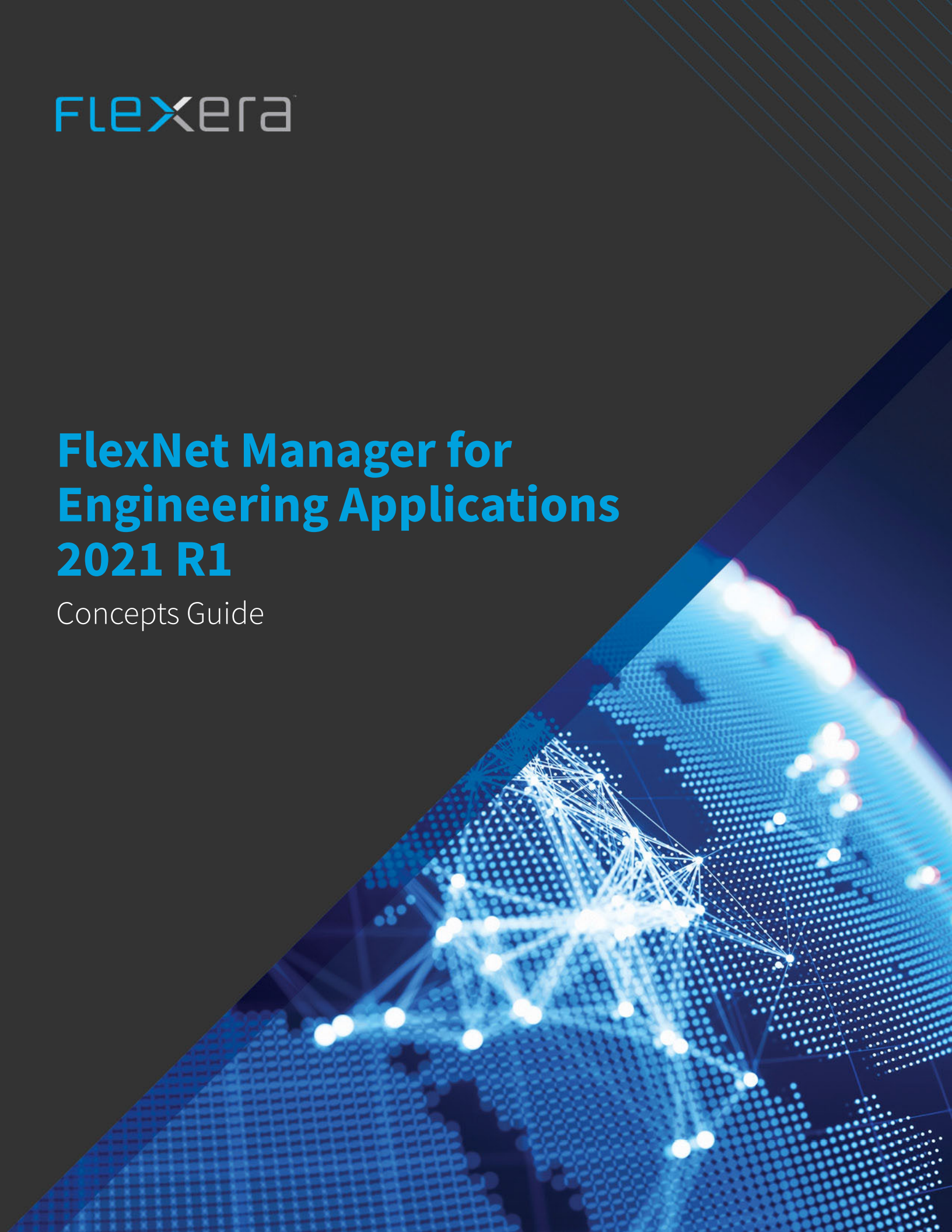




FlexNet Manager for Engineering Applications 2021 R1

Concepts Guide



Legal Information

Book Name: FlexNet Manager for Engineering Applications 2021 R1 Concepts Guide
Part Number: FNM-2021R1-CG01
Product Release Date: July 2021
Document Release Date: July 2021

Copyright Notice

Copyright © 2021 Flexera.

This product contains proprietary and confidential technology, information and creative works owned by Flexera and its licensors, if any. Any use, copying, publication, distribution, display, modification, or transmission of such technology in whole or in part in any form or by any means without the prior express written permission of Flexera Software is strictly prohibited. Except where expressly provided by Flexera Software in writing, possession of this technology shall not be construed to confer any license or rights under any Flexera Software intellectual property rights, whether by estoppel, implication, or otherwise.

All copies of the technology and related information, if allowed by Flexera Software, must display this notice of copyright and ownership in full.

FlexNet Manager for Engineering Applications incorporates software developed by others and redistributed according to license agreements. Copyright notices and licenses for these external libraries are provided in a supplementary document that is available from the Flexera Software download site.

Intellectual Property

For a list of trademarks and patents that are owned by Flexera Software, see <https://www.flexera.com/company/legal/intellectual-property.html>. All other brand and product names mentioned in Flexera Software products, product documentation, and marketing materials are the trademarks and registered trademarks of their respective owners.

Restricted Rights Legend

The Software is commercial computer software. If the user or licensee of the Software is an agency, department, or other entity of the United States Government, the use, duplication, reproduction, release, modification, disclosure, or transfer of the Software, or any related documentation of any kind, including technical data and manuals, is restricted by a license agreement or by the terms of this Agreement in accordance with Federal Acquisition Regulation 12.212 for civilian purposes and Defense Federal Acquisition Regulation Supplement 227.7202 for military purposes. The Software was developed fully at private expense. All other use is prohibited.

Contents

1	Preface	5
	Intended Audience	5
	Additional Documentation	6
2	Overview.....	7
	FlexNet Manager for Engineering Applications Architecture.....	7
	FlexNet Manager for Engineering Applications.....	8
	Deployment Options.....	8
	Getting Started with FlexNet Manager for Engineering Applications	9
	The FlexNet Manager for Engineering Applications User Interface	10
3	License Server Management	11
	License Server User Interface.....	11
	FlexNet Manager for Engineering Applications Roles and Permissions	12
	Starting and Stopping FlexNet License Servers	13
	Stopping License Servers.....	14
	Starting License Servers.....	15
	Stopping Vendor Daemons.....	16
	Starting Vendor Daemons	16
	Determining FlexNet License Server Status.....	17
	Configuring the License Server	17
	Common Vendor Daemon Mapping	19
4	FlexNet License Administration	21
	License Basics.....	21
	License Rights Representation.....	21
	Trusted Storage	22
	Trusted Storage License Groups.....	22

Licenses Tab	23
Activity Tab	23
Interpreting License Counts	23
5 Analyzing FlexNet License Usage	27
Strategies for Analyzing Usage Data	27
Basic License Usage Concepts	28
License Denials	28
Eliminating Spurious Denials	30
Duplicate Grouping Information	31
Explanation	32
Overdraft License Usage	33
Deciding When to Add Additional Licenses	34
Balancing License Load Across Your Enterprise	34
Remixing Licenses Within Features	35
6 Investment Planning	37
Basic Investment Planning Steps	37
Products	38
Adding Products	38
Importing Product Definitions	38
Product Definition XML Format	39
Importing Product Definitions at the Command Line	39
Exporting Product Definitions	40
Adding Features to Products	40
Features	40
Software Producers	40
Associating Vendor Daemons with Software Producers	41
Contracts	41
Contract Pools	41
Product Licenses	41
License Distributions	42
Remixing or Renewing Contract Pools	42
Basic Remix or Renewal Process	42
Importing Contract Data	43
Software Vendors	44
Product Choices	44
Product Aggregation	44
Product Event Creation	45
Product Resolution	45
7 Glossary	47
Index	53

1

Preface

This manual provides strategies and guidelines for managing your licenses and license servers using FlexNet Manager for Engineering Applications.

- [Overview](#), contains an architectural and operational overview of a FlexNet-managed system.
- [License Server Management](#), describes the ways in which you can use FlexNet Manager for Engineering Applications to manage your license servers.
- [FlexNet License Administration](#), explains basic licensing concepts that are useful in interpreting license categories and counts displayed in FlexNet Manager for Engineering Applications.
- [Analyzing FlexNet License Usage](#), introduces ways that you can use usage reports to make FlexNet license-management decisions.
- [Investment Planning](#), provides information about FlexNet Manager for Engineering Applications's investment planning functionality, which enables you to input contract data and map features to products, so you can track usage on the product level.



Important • *FlexNet Manager for Engineering Applications Foundation Edition does not include Flexera Analytics. Flexera Analytics is a separately licensed component that enables business intelligence reporting.*

Intended Audience

This manual is intended for license administrators of FlexEnabled and other concurrently licensed software applications at enterprise organizations. It is assumed that you have an understanding of FlexNet Publisher licensing component configuration and operation.

Additional Documentation

In addition to this concepts guide, Flexera Software provides a number of documentation resources to assist you in installing and configuring FlexNet Agents, understanding licensing concepts, and working with FlexNet Manager for Engineering Applications to monitor and manage license usage throughout your enterprise.

The following PDF documents are available from the **Flexera Software Product and License Center** (<http://flexerasoftware.flexnetoperations.com>).

- **FlexNet Manager for Engineering Applications Installation Guide:** Provides installation and configuration information for the initial and subsequent FlexNet Manager for Engineering Applications installations, available from the Flexera Software download site.
- **FlexNet Manager for Engineering Applications Reporting Guide:** Provides information about generating FlexNet Manager for Engineering Applications Classic reports, use of the Cognos reporting facility by Flexera Analytics, and use of the command-line reporting interface.
- **Flexera Analytics Guide:** Provides information about Flexera Analytics (Cognos) reports and capabilities. FlexNet Manager for Engineering Applications Foundation Edition does not include Flexera Analytics.
- **FlexNet Agent Installation Guide:** Provides installation and configuration information for the initial and subsequent FlexNet Agent installations, available from the Flexera Software download site.
- **FlexNet Manager for Engineering Applications Help Library:** For context-sensitive instructions on how to use FlexNet Manager for Engineering Applications, available by clicking the Help button on each FlexNet Manager for Engineering Applications page.
- **FlexNet Publisher License Administration Guide:** For details related to FlexNet licensing component configuration and operation. This guide describes the setup and administration of a FlexNet licensing system, including starting and stopping license servers, setting up an options file, enabling report logging, and using FlexNet license administration tools. (This guide is part of the FlexNet Publisher documentation set.)

2

Overview

FlexNet Manager for Engineering Applications is a Web-based administration, reporting, and planning tool for usage-based licenses and license servers. It provides tools for comprehensive license management that can help you reduce software costs and increase user productivity while leveraging existing technology.

The key to meeting these objectives is information—knowing what licenses your enterprise is using, knowing which individuals are using them, and knowing how frequently (and for how long) the licenses are being used.

When you have this information, you can do the following:

- Identify areas of overspending.
- Share expensive licenses across your enterprise.
- Reduce license-administration costs.

FlexNet Manager for Engineering Applications provides a single console view of all your FlexNet licenses and license servers, which enables you to centrally manage, track, and control them across your enterprise. You identify your license server environment and immediately have a network-wide view of your enterprise's FlexNet license assets. From the management console, you can monitor and manage license servers, set alerts to identify potential problems, view real-time license usage, and run historical usage reports.

FlexNet Manager for Engineering Applications Architecture

FlexNet Manager for Engineering Applications communicates with a FlexNet license server using a FlexNet Agent that is installed on each license server machine. A single FlexNet Manager for Engineering Applications installation can be used to manage licenses and license servers throughout your organization.

For descriptions of the various deployment options, see the Deployment Options section in the *FlexNet Manager for Engineering Applications 2021 R1* Installation Guide.

FlexNet Manager for Engineering Applications

FlexNet Manager for Engineering Applications enables the centralized management of license servers, licenses, and historical usage data throughout an enterprise. In most cases, one FlexNet Manager for Engineering Applications installation can serve an entire organization. FlexNet Manager for Engineering Applications consists of FlexNet Manager for Engineering Applications Admin—which provides the capability to manage license servers and vendor daemons—and FlexNet Manager for Engineering Applications Reporting—which provides usage reporting functionality.

FlexNet Agent

FlexNet Manager for Engineering Applications interacts with each license server using a FlexNet Agent that is installed on the license server machine. The FlexNet Agent provides secure, remote administration of one or more license servers on that machine on behalf of FlexNet Manager for Engineering Applications.

Flexera Analytics (Cognos)

Flexera Analytics is a separately licensed component that enables you to generate dynamic reports using Cognos. For more information, see the *Flexera Analytics 2020 R1 Guide*.



Important • *FlexNet Manager for Engineering Applications Foundation Edition does not include Flexera Analytics.*

Deployment Options

Depending on your enterprise's needs, you can deploy FlexNet Manager for Engineering Applications in one of a few different ways. [Figure 2-1](#) shows a 2-server deployment that includes Flexera Analytics.

For information about different FlexNet Manager for Engineering Applications deployment options, see the *FlexNet Manager for Engineering Applications 2021 R1 Installation Guide*.

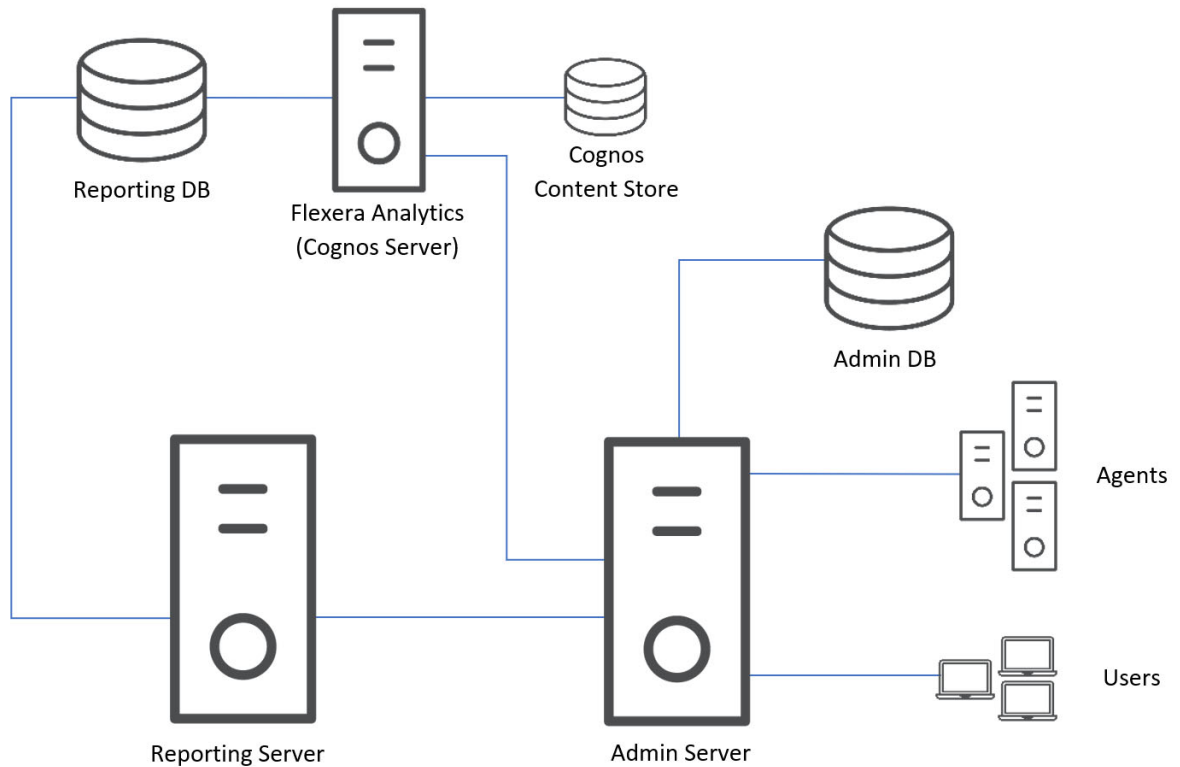


Figure 2-1: Basic FlexNet Manager for Engineering Applications deployment with Flexera Analytics (Cognos)

Getting Started with FlexNet Manager for Engineering Applications

Read this section after you have installed FlexNet Manager for Engineering Applications and tested the management of at least one FlexNet Publisher license server. For installation and testing information, see the *FlexNet Manager for Engineering Applications 2021 R1 Installation Guide* and *FlexNet Agent Installation Guide*.

Prerequisites

To make the most of managing your FlexNet licensing environment using FlexNet Manager for Engineering Applications, you need a thorough understanding of the following:

- Your network topology and host names.
- The names and locations of all the vendor daemons that contribute to your managed environment.
- The location of all FlexNet license servers and license files.
- The location of report logs, debug logs, and options files associated with your license servers.
- The names of the users who are authorized to access FlexNet Manager for Engineering Applications.
- The database, operating system, and browser versions that are supported for use with FlexNet Manager for Engineering Applications 2021 R1. (See the *FlexNet Manager for Engineering Applications 2021 R1 Release Notes*.)

Considerations for Using FlexNet Manager for Engineering Applications

FlexNet Manager for Engineering Applications maintains a database of all details related to license-server topology, management, and status. After you begin using FlexNet Manager for Engineering Applications in your environment, you should not revert to manual license server operations.

The FlexNet Manager for Engineering Applications User Interface

Figure 2-2 shows the Dashboard that is displayed when you sign in to FlexNet Manager for Engineering Applications. The navigation bar across the top of the page provide access to different pieces of functionality within the system.

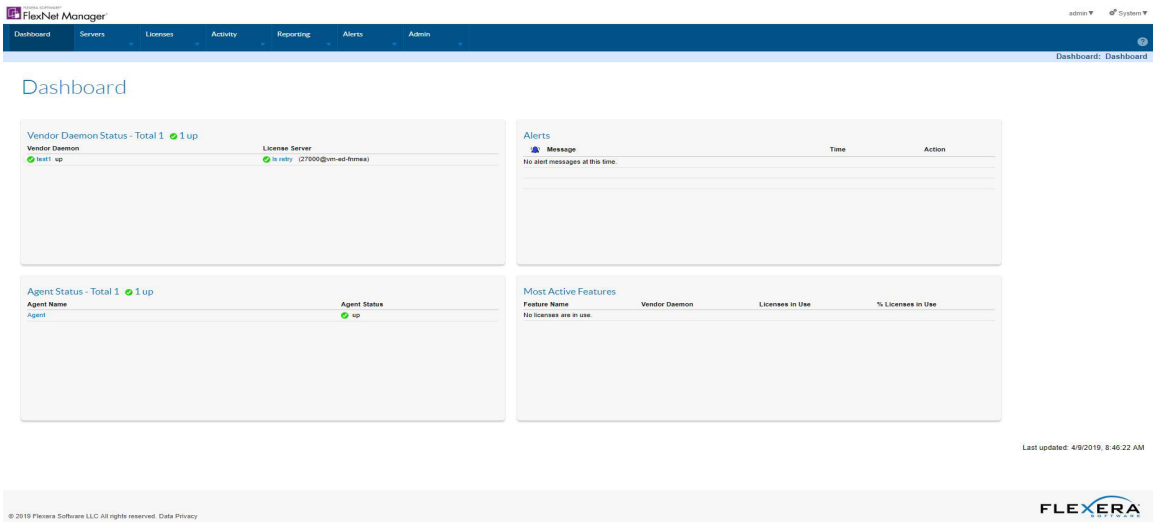


Figure 2-2: FlexNet Manager for Engineering Applications Dashboard

License Server Management

FlexNet Manager for Engineering Applications assists you in the network-wide management of licensing activities at your enterprise. This chapter provides an introduction to license-management concepts and explains how to [map vendor daemons to a common vendor daemon](#) (in cases where a software publisher deploys a single vendor daemon to serve features previously authenticated by one or more vendor daemons).

License Server User Interface

You access license server management using the **Servers** tab on the user interface. This page presents the portal through which you can access all aspects of license server status and management. Clicking a license server name displays in-depth information about that license server.

Figure 3-1 shows the **License Manager Summary** page, which is the main page for managing your license servers.**

License Server	Port@Host	License Manager Status	Vendor Daemon Status	Last Update	
supreme-demoTest	27008@supreme	❌ down	❌ down	Jan 16, 2013 9:45 AM	Update Status
supreme-avned9	65534@supreme	✅ up	✅ up	Jan 16, 2013 9:47 AM	Update Status
supreme-schi8	28024@supreme	✅ up	✅ up	Jan 16, 2013 9:49 AM	Update Status
supreme-tvend10	@supreme	✅ up	✅ up	Jan 16, 2013 9:48 AM	Update Status

Figure 3-1: License Manager Summary page

FlexNet Manager for Engineering Applications Roles and Permissions

Each FlexNet Manager for Engineering Applications registered user has one or more assigned roles. A role specifies the permissions allowed for various operations within FlexNet Manager for Engineering Applications. All roles can be created, edited, and viewed using the **Roles** page (available from the Admin tab). To view the associated permissions and status (active or inactive) for a specific role, click the Role Name.

During installation, the following system-defined roles are created for FlexNet Manager for Engineering Applications. You cannot modify these system-defined roles.

- **License Administrator**—Users with this role can manage licenses and options files.
- **License Server Administrator**—Users with this role can manage license servers and vendor daemons, and make use of all License Administrator permissions.
- **Super Administrator**—This is the most powerful role; it enables users to manage all aspects of FlexNet Manager for Engineering Applications operation.

During FlexNet Manager for Engineering Applications installation, the default user (`admin`) is created and assigned the Super Administrator role.

When planning FlexNet Manager for Engineering Applications deployment, consider creating roles with varying degrees of permissions that correspond with your users' responsibilities. For example, you may want to enable some users to manage license availability, but restrict their ability to start or stop a license server or vendor daemon. To do this, you could create a new role, `ck_usage`, with just the **Manage Licenses** and **Manage Licenses in Use** permissions selected. Then, when you add these users, assign them the `ck_usage` role.

Table 3-1 • FlexNet Manager for Engineering Applications Permissions

Permission	Allowed Activities
Manage License Servers	<ul style="list-style-type: none">● Start license servers● Stop license servers● Reread all license files● Modify server configurations● Schedule activities
Manage Vendor Daemons	<ul style="list-style-type: none">● Start vendor daemons● Stop vendor daemons● Reread the license file for a specific vendor daemon● Modify vendor daemon configuration● Schedule and rotate report logs
Manage License	<ul style="list-style-type: none">● Add new licenses● Delete inactive licenses

Table 3-1 • FlexNet Manager for Engineering Applications Permissions

Permission	Allowed Activities
Manage Licenses in Use	Free a license currently checked out
Manage Options for Vendor Daemons	Add, edit, and delete items in options files

Starting and Stopping FlexNet License Servers



Important • When you begin using FlexNet Manager for Engineering Applications, it must have complete control over the license servers throughout your enterprise. Do not perform server operations manually. Additionally, you should disable all Windows services and UNIX startup scripts that have been configured to automatically manage the FlexNet license manager—*lmadmin* or *lmgrd*—and associated vendor daemons.

The FlexNet license server is a virtual container of processes consisting of one license manager process (*lmadmin* or *lmgrd*) and one or more vendor daemons. All members of the container must reside on the same machine, with one or more license servers existing on a given machine.

FlexNet Manager for Engineering Applications maintains the notion of a license server display name, which can be thought of as a shortcut to refer to the set of processes for a given license server. One or more license servers are configured when a FlexNet Agent is installed.

To connect a FlexNet Agent to FlexNet Manager for Engineering Applications, you must add the FlexNet Agent using the Add Agent page, then connect to the FlexNet Agent. For more information, refer to the FlexNet Manager for Engineering Applications 2021 R1 Help Library (available by clicking the Help button in the user interface).

Figure 3-2 shows an example license server machine with three license servers: *Server_1*, *Server_2*, and *Server_n*, along with its FlexNet Agent.

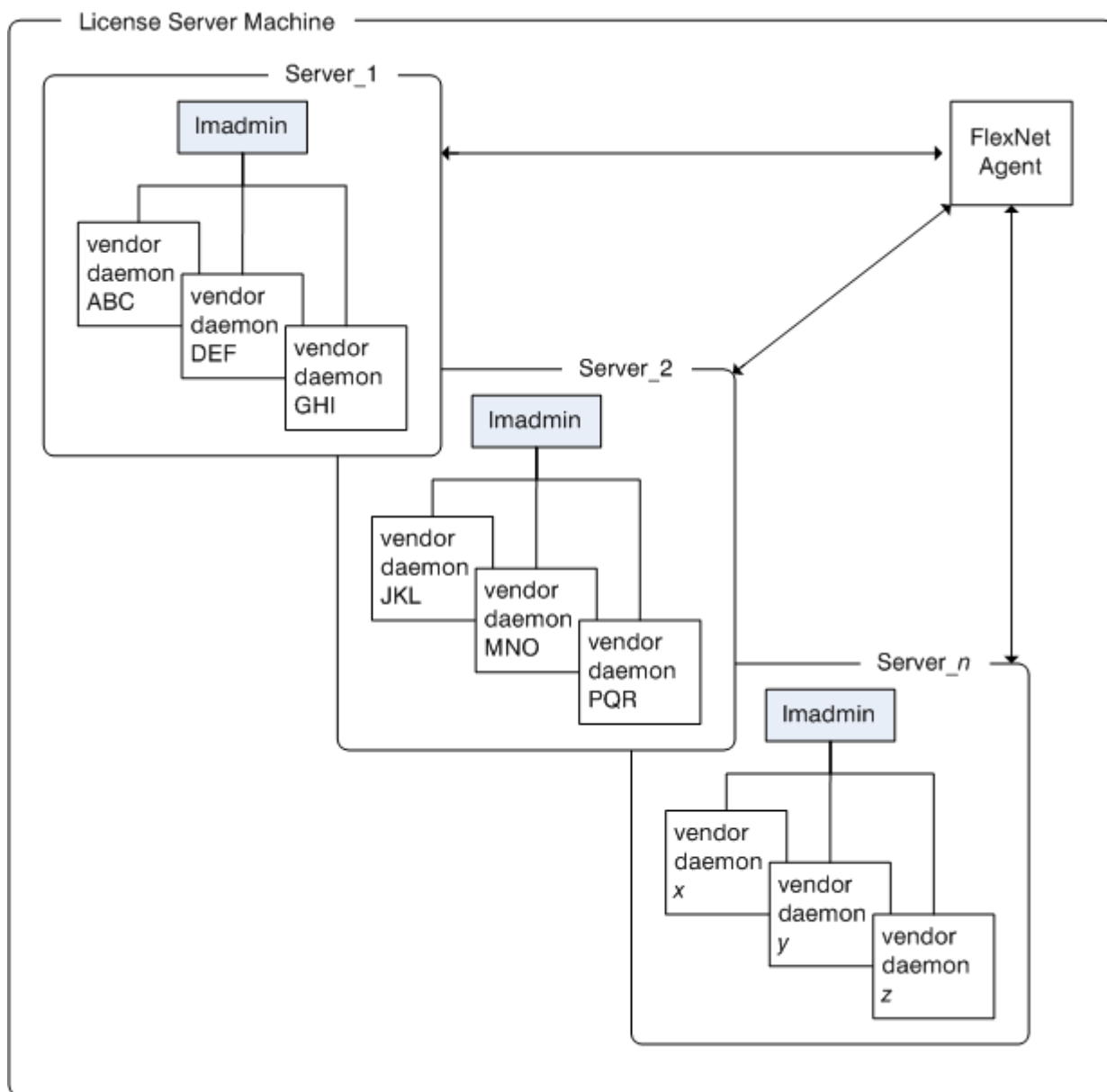


Figure 3-2: Sample FlexNet license server machine

Stopping License Servers

Stopping a license server stops its Imadmin process, as well as all vendor daemon processes managed by that Imadmin process. All licenses served by that license server are checked back in. The connection between the license server and FlexEnabled application which it serves licenses is severed. In most cases, the application exits. (Consult your software vendor for the specific application behavior.) Exercise caution when stopping a license server.

This operation requires the Manage License Servers permission. See [“FlexNet Manager for Engineering Applications Roles and Permissions”](#) for details.

Notifying Users

Before stopping a license server, ensure that users of licenses for the license server are aware of the impending shutdown.



Task

To identify and notify users:

1. Click the **Activity** tab, then click Concurrent **By User**. This displays the Concurrent Usage by User - Summary page.
2. Click the **Host Name** column heading to sort the list by host name and find all entries for the license server display name of interest.
3. Notify all users of the impending license server shutdown.

Stopping the License Server

After all users have checked in the licenses for features served by the vendor daemon, you can stop the license server.



Task

To stop the license server:

1. Click the **Servers** tab, then click **License Managers**.
2. Click the display name for the license server of interest. This displays the **Configure and Manage License Server** page for that license server.
3. Click the **Stop** button.
4. After a period of time—several seconds to several minutes, depending on your network response time—observe the License Server Status on this page. The License Manager Status field and all Vendor Daemon fields should contain the red x icon indicating a successful shutdown.

At this point, the license server of interest, including its `lmgrd` and all vendor daemon processes, are stopped. Other license servers on the same machine are not affected.

Starting License Servers

This operation starts a license server, including its `lmgrd` process and all vendor daemon processes. The user must have the Manage License Servers permission. See “[FlexNet Manager for Engineering Applications Roles and Permissions](#)” for details.



Task

To start a license server:

1. Click the **Servers** tab, then click **License Managers**.
2. Click the display name for the license server of interest. This displays the **Configure and Manage License Server** page for the license server.
3. Click the **Start** button.

After a period of time—several seconds to several minutes, depending on your network response time—the **License Manager Status** and **Vendor Daemons** fields will contain a green check mark icon if the license server was successfully started.

If, after a reasonable amount of time, these fields still contain the red x icon, review the FlexNet application logs to determine why the license server did not start. See the *FlexNet Manager for Engineering Applications 2021 R1 Installation Guide* for more information about application logs.

Stopping Vendor Daemons

This operation requires “Manage Vendor Daemons” permission. See “[FlexNet Manager for Engineering Applications Roles and Permissions](#)” for details.

Notifying Users

Make sure that users of licenses for the vendor daemon are aware of the impending shutdown.



Task

To identify and notify your users:

1. Click the **Activity** tab, then click **Concurrent by Feature** to display the Concurrent by Feature - Summary page
2. Click the Vendor Daemon column to sort the list by vendor daemon name and find all entries for the vendor daemon name of interest.
3. Click the **Users** link for each feature served by the vendor daemon and note the user names.
4. Notify all users of the impending vendor daemon shutdown.
5. Wait for licenses to be checked in, or click the **Free Selected Licenses** button for each feature to check the licenses back in on behalf of the users.

Stopping the Vendor Daemon

When all users have checked in the licenses for features served by the vendor daemon of interest, you can stop the vendor daemon.



Task

To stop the vendor daemon:

1. Click the **Servers** tab, then click **Vendor Daemons**.
2. Click the display name for the vendor daemon of interest. This displays the **Configure and Manage Vendor Daemon** page for the vendor daemon.
3. Click the **Stop Daemon** button.
4. After a period of time—several seconds to several minutes, depending on your network response time—observe the Vendor Daemon Status on this page. This field should contain the red x icon indicating a successful shutdown.

Starting Vendor Daemons

This user who performs this operation must have a role with the Manage Vendor Daemons permission. See “[FlexNet Manager for Engineering Applications Roles and Permissions](#)” for details.



Task

To start a vendor daemon:

1. Click the **Servers** tab, then click **Vendor Daemons**.
2. Click the display name for the vendor daemon of interest. This displays the **Configure and Manage Vendor Daemon** page for the vendor daemon.
3. Click the **Start Daemon** button.
4. After a period of time—several seconds to several minutes, depending on your network response time—observe the Vendor Daemon Status. This field should contain a green check mark icon indicating a successful startup.

If, after a reasonable amount of time, these fields still contain the red x icon, review the FlexNet application logs. See the *FlexNet Manager for Engineering Applications 2021 R1 Installation Guide* for more information about application logs.

Determining FlexNet License Server Status

The **License Manager Summary** page (available by clicking the Servers tab, then clicking License Managers) provides overall status information for each license server within your enterprise, including all `lmgrd` and vendor daemon processes. Status is updated every 5 minutes by default; click the **Update Status** button for current status.

License Manager Status

For detailed status on the particular `lmgrd` process for a given license server, click the license server display name. From here, you can configure and view license server logs, configure schedule details, and perform other management tasks related specifically to the `lmgrd` process.

Vendor Daemon Status

The Vendor Daemons subtab provides status information for each individual vendor daemon in your environment. Click the display name for the vendor daemon of interest. The Configure and Manage Vendor Daemon page is displayed. From here you can configure details related to a vendor daemon's report log, daemon-specific debug log, and the group to which the vendor daemon belongs.

Configuring the License Server

This operation requires Manage License Servers or Manage Vendor Daemons permissions, as appropriate. See “[FlexNet Manager for Engineering Applications Roles and Permissions](#)” for details.

You can configure various aspects of your license servers and their associated `lmgrd` and vendor daemon processes using the **Servers** tab. Table 3-2 lists the configuration items available. Consult the *FlexNet Manager for Engineering Applications Help Library* for details on each item.

Table 3-2 • FlexNet Manager for Engineering Applications Configuration

Configuration Item	How to Configure
Refresh rate for license server status	1. Click the Servers tab, then click License Managers.
License file reread interval	2. Click the display name for the license server of interest to open the Configure and Manager License Server page. 3. Click the License Server Scheduling link to open the License Server Scheduling page. 4. Modify the setting as appropriate. 5. Click Save.
License server display name	1. Click the Servers tab, then click License Managers.
License manager port	2. Click the display name for the server of interest.
License manager options	3. Modify the values in the License Manager Configuration section as appropriate.
Auto Restart	4. Click Save.
License server host name	
Debug logging activation	
License manager debug log location	
Vendor daemon options	1. Click the Servers tab, then click Vendor Daemons. 2. Click the display name for the vendor daemon of interest. 3. Click Edit Options File. 4. Click the Edit button and make modifications as appropriate.
Vendor daemon group name	1. Click the Servers tab, then click Vendor Daemons.
Vendor daemon debug log location	2. Click the display name for the vendor daemon of interest.
Report logging activation	3. Modify the vendor daemon value as appropriate.
Report log name	4. Click Save.
Report log rotation interval	

Common Vendor Daemon Mapping

FlexNet Publisher provides a *common vendor daemon* option to software publishers who want to deploy a single vendor daemon that serves features previously served by other vendor daemons. This scenario typically applies to software publishers who have acquired or merged with other companies, where the individual companies have previously sold FlexEnabled applications with their own vendor daemons.

For example, *ABCware* merges with *DEFware*, and begins to deploy the new vendor daemon *abcdef*. When the *abcdef* vendor daemon registers with FlexNet Manager for Engineering Applications, it receives usage information only about features served by the *abcdef* vendor daemon. Data received about features served by previously deployed *abcware* and *defware* vendor daemons is disregarded. To ensure that FlexNet Manager for Engineering Applications receives all relevant data for the merged publisher's features, you must map the secondary vendor daemon names (*abcware* and *defware*) to the primary vendor daemon name (*abcdef*).

Use of a common vendor daemon enables the publisher to use one vendor daemon to serve features with different vendor names. A common vendor daemon configuration contains:

- One primary vendor daemon—This is the vendor daemon for the acquiring company.
- One or more secondary vendor daemons—These are the vendor daemons that were deployed prior to the merger or acquisition.



Task

To configure common vendor daemon mapping:

1. Stop the Admin server if it is running.
2. Open a command prompt and change to the following directory—
`<fnmea_admin_install_dir>/release/manager/webapps/flexnet/WEB-INF/classes`
3. Open the file `secondaryVendors.properties`. (This is an empty file.)
4. Add the common vendor daemon mappings to the `.properties` file in the following manner:

```
secondaryVendorName1=primaryVendorName
secondaryVendorName2=primaryVendorName
```

In the case of the previous example, the `.properties` file would contain the following:

```
abcware=abcdef
defware=abcdef
```

5. At the command prompt, change to the `<fnmea_admin_install_dir>` directory, then type the following command:
`flexnet site make`
6. Restart the Admin server.

FlexNet License Administration

The ability to view and administer licenses is a valuable tool to assist you in managing your license assets and allocating them efficiently.

The administration capabilities available in FlexNet Manager for Engineering Applications enable you to:

- View your license assets, as well as their usage and availability.
- Manage license rights by adding licenses or freeing inactive licenses.

This chapter explains basic licensing concepts that are useful in interpreting license categories and counts displayed in FlexNet Manager for Engineering Applications.

For information on usage reports, see the *FlexNet Manager for Engineering Applications 2021 R1* Reporting Guide. See [Analyzing FlexNet License Usage](#) for ways in which you can analyze usage data to determine your licensing needs.

License Basics

This section discusses the two main types of license rights representation and the three main groups of licenses and their organization in FlexNet Manager for Engineering Applications.

License Rights Representation

FlexNet Manager for Engineering Applications supports the management of licenses represented in license files and in *trusted storage*.

License Files

A license file is a text file containing license certificates from one or more software publishers. A license certificate, in turn, is licensing rights information specific to a single publisher's FlexEnabled application.

Trusted Storage

Trusted storage is a secure location in which a software publisher can install license rights—in the form of fulfillment records. Unlike a license file, which cannot be changed by the publisher after it is delivered to an end user, license rights in trusted storage can be updated by the publisher.

Also, license rights in trusted storage can be transferred (rehosted) from one license server to another within an end-user enterprise without the involvement of the software publisher.

License rights in trusted storage can be equivalent to those expressed in license files. Just as license files can be placed on both end-user machines and license-server machines, trusted storage can be created on end-user machines and license-server machines.

Trusted Storage

Trusted storage is a secure storage area used to store fulfillment records and other licensing information.

Fulfillment records in trusted storage contain four types of information:

- License lines: Similar to FEATURE and INCREMENT lines in license files, but containing values that may be overridden by other data held in the fulfillment record
- Override values for the license lines in the fulfillment
- Details of the transaction that provided the fulfillment record
- Details of how the licenses can be used by the server—as concurrent licenses, as *activatable* licenses (see “[Trusted Storage License Groups](#)”), and for transfer to other servers.

Trusted storage is segmented into publisher-specific areas, each identified by a Trusted ID. The security of each segment is set separately, using different binding and anchoring configurations. Anchors detect deletion, tampering, and subsequent restoration of trusted storage. Binding is the combining of one or more properties of a machine into a unique fingerprint, which is used to lock trusted storage to that machine.

Trusted Storage License Groups

License rights in the license server’s trusted storage can be served to FlexEnabled products on end-user machines or transferred to other license servers. These licenses are divided into three groups. The distribution of licenses into these groups is determined by the software publisher whose FlexEnabled products you license.

- **Concurrent licenses:** License rights available network-wide from a license server system that can be shared by a predetermined number of simultaneous instances of the FlexEnabled application. These are similar to the license rights within a license file served by a license server.

Concurrent licenses can be stored in both license files and trusted storage.

- **Detachable licenses:** License rights that serve a dual purpose; that is, when a system is on the network and can access a license server, detachable license rights function as concurrent license rights.

However, these licenses can also be securely detached onto an end-user machine for a certain amount of time. This means that detachable licenses can be used to run an application on a computer that is no longer connected to the license server, such as a laptop being used at home.

- **Activatable licenses:** License rights that have been delivered to a license server system, but not yet made available for checkout—after being activated on an end-user machine, the rights are available for checkout by the FlexEnabled application on that end-user machine.

Activatable licenses are stored only in trusted storage, not in license files. Counts for activatable licenses are in terms of products, not features.

Licenses Tab

The Licenses tab displays deployed licenses by type—concurrent, detachable, activatable; and by license rights representation—license files, fulfillment records in trusted storage.

Concurrent license information can come from license rights in both license files as well as trusted storage. The Concurrent Licenses page provides the following information for each concurrent license:

- Vendor daemon
- Feature name
- Feature version
- Count information, including overdraft counts (see [Interpreting License Counts](#))
- A link to access information on servers serving the feature

Activity Tab

Real-time license status is displayed under the Activity tab for concurrent, activatable, and detachable licenses. You can view concurrent license activity by feature or by user, and detachable and activatable license activity by product or by host.

Concurrent licenses are represented in both license files and trusted storage. The View Activity: Concurrent by Feature page lists concurrent license activity as follows:

- Vendor daemon
- Feature name
- Feature version
- Total number of concurrent licenses available for the feature (See [Interpreting License Counts](#))
- Percentage of those licenses currently in use
- Grouping of those licenses into pools (On each license server, licenses for the each feature are divided into pools based on several license characteristics. See the *FlexNet Publisher License Administration Guide*—part of the FlexNet Publisher documentation set—for details.)

Interpreting License Counts

FlexNet Manager for Engineering Applications displays the number of deployed licenses for each license group: concurrent, detachable, or activatable, with and without the overdraft count.

Overdraft Licenses

Overdraft licenses are granted by a software publisher and allow for enterprises to use more licenses than purchased. Usually, these overdraft licenses are valid only for a short period of time before additional licenses must be purchased.

Concurrent Licenses						
Vendor Daemon ▲	Feature Name	Feature Version	Total Count	Concurrent	Overdraft	Details
avend9	300	5.100	24	24	0	Servers
avend9	netf1avend9	1.0	10	10	0	Servers
avend9	netf2avend9	1.0	4	1	3	Servers
avend9	netf4avend9	1.0	1	1	0	Servers
avend9	netf5avend9	1.0	2	2	0	Servers
avend9	netf6avend9	2.0	1	1	0	Servers
avend9	netf6avend9	1.0	1	1	0	Servers
avend9	netf7avend9	1.0	1	1	0	Servers
avend9	netf7avend9	2.0	1	1	0	Servers

Figure 4-1: Concurrent Licenses page

In [Figure 4-1](#), the feature netf2avend9 has a total license count of 4 and an overdraft license count of 3. This means that one license of the four available has been purchased by the enterprise, while the publisher has authorized the use of three overdraft licenses.



Note • FlexNet Manager for Engineering Applications supports triggering an alert every time an activatable overdraft license is used.

Total

Total license counts for a feature or product are computed based on the license source (license file or fulfillment record in trusted storage).

For licenses in license files, the total is the sum of licenses purchased plus overdraft licenses.

$Total = Count + Overdraft$

- Count—The count field of the license line.
- Overdraft—The value set for overdraft in the license line.

For licenses in fulfillment records, the count in the license line is multiplied by the number of licenses to arrive at the total. For example, the total concurrent count for product A is:

$Total = (Count \times Concurrent) + (Overdraft \times ConcurrentOverdraft)$

- Count—The count field of the license line.
- Concurrent—The number of concurrent licenses purchased.
- ConcurrentOverdraft—The number of concurrent overdraft licenses purchased.

To further understand counts as they appear in different pages of your Licenses tab, consider the following example of the license assets of Company A.

Table 4-1 • License Assets for Company A

License Source	Product	Product Count	Feature	Feature Count	License Type
Fulfillment Record 1	P1	6	f1	1	Activatable
			f2	1	
Fulfillment Record 2	P1	10	f1	1	Detachable
			f2	1	
Fulfillment Record 3	P2	1	f1	4	Concurrent
License Certificate 1	-	-	f1	7	Concurrent
	-	-	f2	7	Concurrent

There are several points to note:

- In the column License Source, Company A has both trusted storage, as well as license file–based licenses. A fulfillment record is stored in trusted storage, while license certificates make up license files.
- In the column License Type, trusted storage has licenses of the activatable, detachable, and concurrent types. License certificates only have concurrent licenses. This is always true of license file–based licenses, not just for Company A.
- Features are the basic unit of licensing for both license file- and trusted storage-based licenses. However, in license certificates, features are not grouped into products. (See the Product column in the table above.) For example, both Fulfillment Record 1 and License Certificate 1 contain licenses for features f1 and f2. However, f1 and f2 are grouped to form the product P1 in Fulfillment Record 1.
- Fulfillment Record 1 contains 6 activatable licenses (see Product Count) for product P1.
- Fulfillment Record 2 contains 10 detachable licenses (see Product Count)) for product P1.
- Fulfillment Record 3 contains 1 concurrent license (see Product Count)) for product P2.
- License Certificate 1 contains 7 concurrent licenses each (see Feature Count) for features f1 and f2. These features do not combine to form a product.
- Each product P1 contains one each of features f1 and f2.
- Each product P2 contains 4 of feature f1.

This is how the information appears in the **Licenses** tab for Company A:

- There are a total of 21 concurrent licenses for feature f1:
 $10 \text{ (Fulfillment Record 2)} + 4 \text{ (Fulfillment Record 3)} + 7 \text{ (License Certificate 1)} = 21$

Note that the 6 licenses in Fulfillment Record 1 do not contribute to this total. This is because these are licenses that cannot be checked out of and in to a license server machine; they can only be activated on end-user machines.

- Feature f1 cannot be activated or detached independently; along with feature f2, it combines to form Product P1, for which Company A has activatable and detachable licenses. (See Fulfillment Record 1 and Fulfillment Record 2)
- Checking out a concurrent license for feature f1 does not require checking out a license for feature f2.

Analyzing FlexNet License Usage

This chapter introduces ways in which you can analyze output from the FlexNet Manager for Engineering Applications reporting module in order to understand license-usage patterns in your enterprise.

- [Strategies for Analyzing Usage Data](#)
- [Basic License Usage Concepts](#)
- [Deciding When to Add Additional Licenses](#)
- [Balancing License Load Across Your Enterprise](#)
- [Remixing Licenses Within Features](#)

Strategies for Analyzing Usage Data

Analyze your usage data with the following questions in mind:

- **[Are all users getting licenses when they need them?](#)**

You determine the level of need based on the nature of your FlexEnabled applications. Does a user need one license continuously for eight hours or for just a few seconds? See [“Deciding When to Add Additional Licenses,”](#) and [“License Denials,”](#) for additional information.
- **[What is your usage profile?](#)**

Determine which users show the most license usage activity. Consider reserving licenses just for them and leaving the remaining available licenses for the more casual users.
- **[Are there licenses that are underutilized?](#)**

If so, consider remixing your features. See [“Remixing Licenses Within Features,”](#) for further information.
- **[What is the future staffing in your organization?](#)**

Consider defining groups of users based on categories such as assignment, server usage, or license need. Set up FlexNet licensing groups to help manage license usage.
- **[Is license usage balanced across your enterprise?](#)**

You may have your licenses divided up among the license servers in your enterprise. First, ensure that all license servers are available and serving licenses. Second, make sure that license requests are distributed evenly among your license servers.

If there is one license server that seems to be getting more requests than the rest, consider redistributing the total pool of licenses so that more are available on that license server. See [“Balancing License Load Across Your Enterprise,”](#) for more information.

- **How often should you take a usage snapshot?**

A monthly review of usage data provides you with a helpful view into the usage patterns across your enterprise. You may need to vary the frequency for various reasons; for example, while you are fine-tuning your license usage. Perform an in-depth annual review to determine your license renewal needs.

- **What is the licensing business model your vendor applies to your licenses?**

Take into consideration the renewal or license usage review period put in place by your vendor. Negotiate a contract that incorporates license rehosting and feature remixing.

Basic License Usage Concepts

This section describes basic license usage concepts that will assist you in better understanding how your organization uses licenses and how you can best manage that usage.

License Denials

You can use FlexNet Manager for Engineering Applications to analyze license denials. A denial indicates that the software was not able to check out a license at the time that the license was requested.

The Usage Summary report gives you an idea of the number of denials relative to the total number of license requests. If you want to determine specific details about each license denial, generate a Raw report specifying the Reason for Denial/Checkin field in Step 5 of the Create Report Configuration wizard for Raw reports.

The following table provides possible explanations for license denials in Raw report output.

Table 5-1 • Denials in Raw report output

Denial	Explanation
All licenses are reserved for others.	The available licenses are being withheld because some of the licenses are reserved for other users in the options file.
Checkout exceeds MAX specified in options file.	The available licenses are being withheld because there is a MAX available for this feature set in the options file.
Checkout request rejected by vendor-defined checkout filter.	The application has a built-in test for the license that failed and rejected the license checkout.
Clock difference too large between client and server.	The license checkout request was denied because the time difference between the client and server was more than 24 hours.

Table 5-1 • Denials in Raw report output

Denial	Explanation
Duplicate selection mismatch for this feature.	A different version of the application is in use and has specified Duplicate Grouping criteria that is different than the one specified in the checkout request.
Feature has expired.	The license for this feature has expired; a new license is required.
Feature removed during lmreread or wrong SERVER line hostid.	Checkout failure due to two possible causes: <ul style="list-style-type: none"> • The feature is removed during lmreread, but the FlexEnabled application is reading an old copy of the license file that still has the removed feature. • The hostid on the SERVER line is for a different host, so all features in this license file were removed.
Feature start date is in the future.	The feature is not enabled yet. The current date is before the feature start date.
FlexNet Publisher version of vendor daemon is too old.	The vendor daemon is built with an older version of FlexNet Publisher than the client application. The vendor daemon must use at least the same version of FlexNet Publisher as the client application. Contact the software vendor for a new vendor daemon.
License BORROW support not enabled.	A user requested license borrowing, but the license or the application does not support license borrowing.
License file does not support this version.	The version of the application is not supported by the license. You need to obtain a new license from the software vendor.
Licensed number of users already reached.	All available licenses have been checked out.
License server does not support this version of this feature.	The version of the feature is not supported. You need to obtain a new license from the software vendor.
License server system does not support this feature.	The feature has expired on the server, has not yet started, or the version is greater than the highest supported version.
No licenses available to borrow.	There is no license available for the borrower to check out. This applies to applications created using FlexNet Publisher 8 or later.
No such feature exists.	The feature could not be found in the license file.
Platform not enabled.	The software is running on a platform for which it is not authorized by Flexera Software. Contact your vendor for further information.
Request for more licenses than this feature supports.	A checkout request was made for more licenses than are available.

Table 5-1 • Denials in Raw report output

Denial	Explanation
Requested borrow interval extends beyond license expiration date.	The user has specified a borrow period longer than the license allows.
This platform not authorized by license.	The license includes a PLATFORM=keyword and this platform is not listed.
User/host on EXCLUDE list for feature.	The user is restricted from accessing a license because it is listed in the EXCLUDE list in the options file.
User/host not on INCLUDE list for feature.	The user is restricted from accessing a license because it is not listed in the INCLUDE list in the options file.
USER_BASED license has not specified users -- see server log.	The user-based licenses must have users specified in the options file.
Users are queued for this feature.	All available licenses have been checked out and there are other users in the queue for this feature.

If a client application was built using FlexNet Publisher 9.0 or later, raw report output displays ultimate denial events separately from other denial events. Only true (ultimate) denial events by vendor daemon will be in the Usage Summary report. All that are not ultimate denial events are ignored.

Eliminating Spurious Denials

If an application requests a license from a license server that uses a license file containing several INCREMENT lines for the same feature, it is possible to get a cluster of denials reported before a successful checkout is obtained.

You can eliminate extraneous or *spurious* denial events from a report by selecting the **Eliminate spurious denials** check box during the Filter Properties step of the Create Report Configuration wizard and specifying a time interval in seconds. With this filter set, denials for the same feature, user, host, process id, and report log file that occur in clusters within the specified time interval are considered qualified for elimination.

FlexNet Manager for Engineering Applications analyzes the qualified denials in each report log according to the following criteria and eliminates extraneous denials from a report:

- If the FlexEnabled application is built with a version of FlexNet Publisher earlier than version 9.0, all qualified denials occurring within the time interval are considered together. If at the end of the time interval, the license has not been granted, the last denial is reported and all previous denials in the interval are eliminated from the report. If a series of denials is seen followed by a checkout, the checkout is reported and the denials are eliminated.
- If the FlexEnabled application is built with FlexNet Publisher 9.0 or later:
 - An ultimate denial event is never folded.
 - Regular denials are always ignored.

Reports that display denial information contain several lines in the report header that indicate whether denial filtering is turned on and if so, show the results of the denial filtering. The results include the number and the percentage of eliminated denials.

The specified time interval should not be more than a few seconds. A reasonable interval corresponds to the number of seconds an individual license server spends granting a license. It should not be the total amount of time the application might wait for a license to be granted—that total time may include network latency and waiting for one or more license servers to respond to multiple requests. Large time intervals may impact FlexNet Manager for Engineering Applications performance because more events are queued to analyze the denials.

Note that this filter considers eliminating only clusters of denials reported in one report log, that is, only clusters of denials issued by the same license server. It will not eliminate “false” denials that are the result of checkout attempts from several license servers in a license file list.

Duplicate Grouping Information

Several fields can be added to Raw report output to help correlate checkouts related by duplicate grouping: `handle`, `shared_handle`, and `shared_group`. All of these fields are identifiers that are re-used by a vendor daemon, and are not, therefore, unique.

All license checkouts are assigned a handle. If a checkout shares a license with a previous checkout, as a result of duplicate grouping, the later checkout is assigned a shared handle that matches the previous checkout’s handle.

All license checkouts are assigned a shared group identifier, in addition to a handle. All checkouts that share the same license have the same shared group identifier.

A shared handle is meaningful on a checkout, not necessarily on both a checkout and its associated checkin. For example, consider an application license that has duplicate grouping based on host.

- If user A starts the application on host A and checks out a license, that checkout has a handle and a shared group identifier.
- When user B starts the application on the same host, user B’s application instance shares a license with user A’s application instance. User B’s checkout has its own handle, a shared handle that matches user A’s checkout handle, and the same shared group as user A’s checkout.
- If user C starts the application on host A and checks out a license after user A has checked in his license, but while user B still has his license checked out, user C’s application instance shares a license with user B’s application instance. User C’s checkout has its own handle, a shared handle that matches user B’s handle, and the same shared group as user B’s checkout.

Note that the checkouts of user A and user C each overlap with the checkout of user B and are all, therefore, part of the same legitimate shared group. Therefore, be careful filtering out usage when examining duplicate grouping information—you might filter out a checkout that links the usage of legitimate shared groups and the filtered data might incorrectly indicate a re-used shared group handle rather than a legitimate shared group.

Time 1	User A checks out license on Host A	A: handle 1 A: shared group 100
Time 2	User B checks out license on Host A	B: handle 2 B: shared handle 1 B: shared group 100
Time 3	User A checks in license on Host A	

Time 4 User C checks out license on Host A

C: handle 3

C: shared handle 2

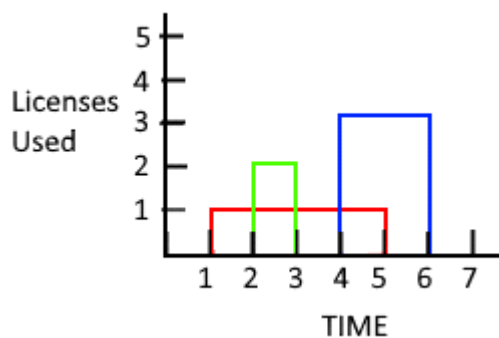
C: shared group 100

Explanation

Consider the following grouped events:

- Time=1: Checkout licCount=1 Group handle=101
- Time=2: Checkout licCount=2 Group handle=102
- Time=3: Checkin licCount=2 Group handle=102
- Time=4: Checkout licCount=3 Group handle=103
- Time=5: Checkin licCount=1 Group handle=101
- Time=6: Checkin licCount=3 Group handle=103

This can be visualized like so:



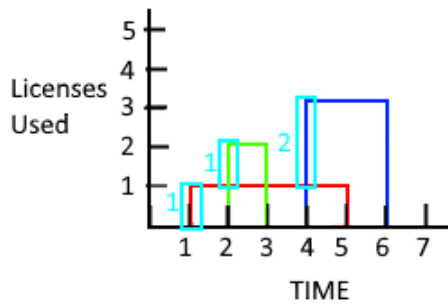
The part that were causing confusion were the checkouts at 2 and 4.

At 2, we were seeing a checkout for 2, and thinking “usage should increase by 2”, however, since one license is shared, it was actually only incrementing by 1.

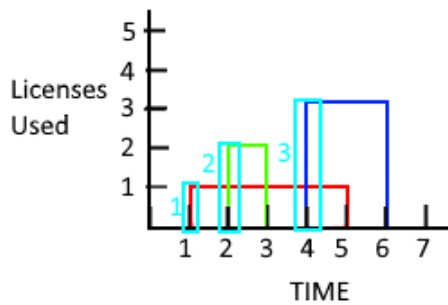
At 4, we see a checkout, and had thought “usage should increase by 3”, but again because one license is shared, we only see usage of 2 at that point.

So, even though we have three checkouts for 1, 2, and 3 licenses, we only see a total usage of 4, not 6.

le: we count usage like this



Not like this:



Overdraft License Usage

Your vendor may have issued you some overdraft licenses in addition to the regular licenses. The overdraft licenses for a feature include the `OVERDRAFT=overdraft_count` keyword in a FEATURE line. FlexNet Manager for Engineering Applications reports on the availability and usage of overdraft licenses differently depending on the type of report.

In a Usage Over Time or High Water Mark report, the number of available licenses, the total number of available and overdraft licenses, and the number of licenses used are shown separately. Clearing the **Show Available Licenses** check box in Step 5 of the Create Report Configurations wizard for a Usage Over Time or a High Water Mark report will suppress both the line representing license availability and the line representing total license availability.

In a Usage Efficiency report, overdraft usage is denoted with an asterisk, “*,” next to the bar. Overdraft usage is also shown in the Type column of RIF output for a Usage Efficiency report. The Type column appears in RIF output only if there are overdraft licenses available or unlicensed usage.

In a Summary Barchart report, a Usage Summary report, or a Raw report, maximum availability and usage of available and overdraft licenses are not distinguished.

Deciding When to Add Additional Licenses

By analyzing the output from a combination of reports, you can determine if you have enough licenses for the users in your enterprise. The first step is to determine if all users can gain access to software when they need it. The following scenario demonstrates a method for making this determination.

1. Generate a Usage Over Time report, sorted by feature.

This report shows the actual number of licenses in use for each feature specified in the configuration during any given period of time versus the total available number of licenses. The granularity of this report is controlled by the Summary Period setting in the configuration. Note any peaks in usage when all available licenses are in use. This may indicate the possibility that some users were denied a license.

2. Generate a Usage Summary report, sorted by feature and then user.

This report breaks down the usage for each feature so you can see who is using the licenses. Locate data for the feature in question from [Step 1](#). Review the usage statistics for each user to determine the quantity of license request denials. If any users show denials, you should investigate further; see [License Denials](#) for further information.

3. Generate a Usage Efficiency report, sorted by feature.

This report shows the hours each available license was used for each feature specified in the configuration. Locate the graph for the feature in question. Look for trends such as the number of licenses used to their full capacity and the overall number of licenses used.

Next, use these statistics to make decisions about the number of licenses you need. If the reports indicate daily maximum usage peaks combined with full-capacity efficiency, consider obtaining more licenses from your vendor.

In addition, you should consider the following:

- Your enterprise's present and future staffing requirements, including whether your enterprise may merge with or acquire another organization.
- The duration for which each license is needed. Fewer licenses are needed if they are checked out for a short amount of time.
- Requirements for specific users. If one or two users need dedicated licenses with the balance shared among the rest of the users, consider reserving licenses for the dedicated users and re-evaluating the number you need for the rest of the enterprise.

Balancing License Load Across Your Enterprise

By analyzing license usage on a per-server basis, you can determine if your licenses are distributed across your enterprise in the most efficient manner. For example, usage reports may reveal an unacceptable number of license denials on one license server while, for the same feature, another license server has a surplus of licenses.

The process of redistributing licenses to minimize denials is known as load balancing. Follow these steps to determine which licenses need load balancing:

1. Generate a separate Usage Over Time report for each license server.

This report shows, on a server-by-server basis, which features are at or are surpassing available license usage. Note these feature names when analyzing subsequent reports.

2. As supporting documentation, generate a Usage Summary report, specifying denied event types only, for those license servers reaching peak usage.

This report helps you determine the degree to which denials are happening on a given license server. Look at the data for those features identified in [Step 1](#). If the data indicates a high percentage of denials, consider adding more licenses for the feature to this license server.

3. To further investigate the reason for denials, generate a Raw report, specifying denied event types only and include the Reason for Denial/Checkin field, for those license servers and features reaching peak usage.

This report produces information for each license request denial including its reason. Refer to “[License Denials](#),” for explanations of the different reasons for denials.

The data these two reports provide in combination helps you to determine which license servers in your enterprise have the heaviest license usage. Take some licenses away from over-utilized servers and add them to the under-utilized ones. If your licenses are node-locked, they need to be re-hosted, which will require your vendor’s assistance.

Remixing Licenses Within Features

When you analyze your license usage, you may see that a few features are used more often and generate the greater proportion of license request denials, and that other features are relatively unused. You may consider *remixing* the particular quantity of licenses you have for each feature.

Remixing involves trading in licenses for rarely used or unused features and replacing them with licenses for overutilized features. This process requires the cooperation of your vendor.

To see a profile of feature usage, generate a Usage Summary report sorted by feature. If you see that Feature1 is in use for only a small percentage of the time, compared to Feature2, which has a higher level of usage, you may want to consider remixing the quantity of licenses for those two features.

Investment Planning

FlexNet Manager for Engineering Applications's investment planning functionality enables your enterprise to analyze product usage trends. You can use this information to forecast future product license needs and product license distribution for the entire organization.

Using the investment planning functionality available from the Planner tab, you map licensed features to the software products that are used in your organization. This feature-to-product mapping enables FlexNet Manager for Engineering Applications to report on product usage, in addition to feature usage. Using the product-usage reports, you can identify usage trends, and use this information to optimize your product license purchases by [remixing](#) your contract pools according to the terms specified in your license contract.

Key investment-planning features include the following:

- The ability to input and manage contractual data.
- Feature-to-product mapping to facilitate product-based usage reporting.
- Remix calculations and analysis to show where license re-allocation may benefit your enterprise.
- Combined with organizational-structure functionality, investment planning enables you to accurately calculate chargebacks for different departments and project groups within your enterprise. (See the *FlexNet Manager for Engineering Applications 2021 R1 Reporting Guide* and the online help for information about organizational structures.)

Basic Investment Planning Steps

To compare your enterprise's contract (licensing) data to actual product usage, you must complete the following tasks.

1. Add information about software producers (companies that create software products) and software vendors (companies that sell software products).



Note • The software producer and the software vendor may be the same company, if your organization purchased the software product directly from the software producer.

2. Add software products to FlexNet Manager for Engineering Applications, then associate licensed features with the products.

3. Input contract data.
4. Create organizational structures, if necessary. See the *FlexNet Manager for Engineering Applications 2021 R1 Reporting Guide* for information about organizational structures.
5. Run data aggregation. See the *FlexNet Manager for Engineering Applications 2021 R1 Reporting Guide* for information about data aggregation.
6. Run reports to see product-based usage trends.
7. Compare the usage trends shown in the reports to the entitlements specified in your contracts to determine whether you are under- or over-licensing specific software products in your enterprise.
8. Run remixes to determine where your enterprise may be able to optimize licensing costs by purchasing licenses for a different mix of products.

Products

The Products page, available from the Planner tab, displays a list of the products that have been [added](#) to and/or [imported](#) into FlexNet Manager for Engineering Applications.

A software product is composed of one or more licensed [features](#). A [software producer](#) creates a product by grouping together separately licensed features and distributing the products to its enterprise customers, along with the feature licenses.

The same feature can be used in multiple products. This practice enables a software producer to create different product tiers, where each tier contains progressively more (and/or more advanced) features. For example, a software producer may have a professional and an express version of a product, where both products include the same core features, but the professional version includes additional advanced features that are not included with the express version.

To enable product-based reporting in FlexNet Manager for Engineering Applications, you must [add products](#), then [map those products](#) to licensed features.

Adding Products

Before you can map licensed features to a product (to provide the basis for product-based reporting), you create a product definition in FlexNet Manager for Engineering Applications by using the Add Product page in the interface or by importing product definitions using an XML file.

After you add a product, you [assign features](#) to the product to provide the basis for product-level usage reporting.

Importing Product Definitions

From the Products page, you can import product definitions contained in an XML file to populate the FlexNet Manager for Engineering Applications database. You can also import product definitions from the [command line](#). The imported products will be displayed on the Products page.

The XML file must contain product definitions from a single [software producer](#). If the XML file contains a reference to a [software producer](#) that does not already exist in the FlexNet Manager for Engineering Applications system, FlexNet Manager for Engineering Applications creates the software producer.

See [Product Definition XML Format](#) for information about the XML file format and the location of .xml and .xsd sample files.

Product Definition XML Format

The following XML example is from the sample file provided. The [product](#) and [feature](#) settings are the same as the settings presented in the FlexNet Manager for Engineering Applications user interface for the Add Product and Add Product Feature pages.

```
<producer category="EDA Tools" name="Producer ABC" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="products.xsd">
  <product category="PCB Layout" family="Board Designer" name="Board Designer Expert" number="100002"
    version="1">
    <feature daemonName="abcd" name="BoardDes" licenseSystem="flex" licensesPerProduct="1"
      minimumVersion=""/>
    <feature daemonName="abcd" name="layoutPlus" licenseSystem="flex" licensesPerProduct="1"
      minimumVersion=""/>
    <feature daemonName="abcd" name="floorPlanner" licenseSystem="flex" licensesPerProduct="1"
      minimumVersion=""/>
  </product>
</producer>
```

All settings that are required in the user interface are also required in the XML file, except for the following feature settings:

- The licenseSystem attribute is optional. Valid values are “flex” or “lum”; the default value is “flex”.
- The licensesPerProduct attribute is optional. The default value is “1”.

Product Definition Sample Files

To assist you in creating product definition files for import, FlexNet Manager for Engineering Applications provides the following sample files in the FlexNet Manager for Engineering Applications Reporting installation directory (in \site\samples):

- [ProductSample.xml](#)—Contains the format for product definition import with example data.
- [products.xsd](#)—Contains the formal definition that can be used by XML editors to ensure the XML files are correct.

Importing Product Definitions at the Command Line

You can import product definition XML files from the command line.



Task

To import the product definition XML file:

1. From a Command prompt, change to the FlexNet Manager for Engineering Applications Admin installation directory.
2. Type the following command:

flexnet importproducts -u<user> -p<password> -f<filename_with_path> [-s<server>]

Exporting Product Definitions

From the [Products](#) page, you can export product definitions to a .csv (comma-separated values) file. Exporting this data may be helpful if you want to work with your product data using another tool (for example, a spreadsheet).

Adding Features to Products

A feature is a discrete, separately licensed unit of functionality. A single feature can be a part of multiple products. FlexNet Manager for Engineering Applications tracks usage by feature. To enable usage reporting by product (which are composed of one or more features), you must associate licensed features with products.

You can also use the [Assign Features](#) page to assign unassigned features to products.

Features

Within software products, features are discrete, separately licensable units of functionality. A software product is composed of one or more licensed features. To facilitate product-based usage reporting, you must associate licensed features with software products that are used in your enterprise.

The Features page displays a list of the licensed features that have been mapped to [products](#). This feature-to-product mapping facilitates reporting on product usage.

The [Assign Features](#) page displays a list of unassigned features, which are features that have not been mapped to a product.

Software Producers

In FlexNet Manager for Engineering Applications, a software producer is a company that makes software products that use a licensing system to control access to product functionality. The product functionality is defined in the product's features, which are licensed to your enterprise by the software vendor. The licensing terms are defined in a contract.

The software producer may also be the [software vendor](#)—that is, the same company may both produce and sell its software products to customers.



Note • *LinkRight Tracker used the term vendor to refer to both the company that produces software and the company that sells software. In FlexNet Manager for Engineering Applications, these two companies are represented by software producers and software vendors, respectively.*

Each product that you define in FlexNet Manager for Engineering Applications must be associated with an existing software producer.

Software Producer Contacts

To help you better manage your relationship with software producers, you can define contact information for various personnel within the software producer's company. For example, you might add contact information for the support representatives who service your account.

Associating Vendor Daemons with Software Producers

The vendor daemon is the process that serves licenses for a particular product's features. A single software producer may have multiple vendor daemons associated with it.

Vendor daemons are automatically added to the FlexNet Manager for Engineering Applications database as needed, when report logs are imported. You associate vendor daemons with software producers to tie the usage of features—served by vendor daemons—to the usage of products, which are created by software producers.

Depending on whether the vendor daemon has already exists in the FlexNet Manager for Engineering Applications database, you can:

- Assign an unassigned vendor daemon to a software producer.
- Manually add a vendor daemon to a software producer.

Contracts

A contract defines the terms under which you (the enterprise) purchase product licenses from a [software vendor](#) (which may be the same company as the [software producer](#)). The contract terms include the start and end dates, the fixed and variable costs of the contract, any license restrictions (for example, geographic license restrictions), and the [remix rules](#) for the contract, which define the percentage of the contract for which you can reallocate licenses.

The product licenses purchased under a contract are typically grouped together into one or more contract pools.

The Contract Summary page, available under the Planner tab, shows a list of the contracts that have been input to or imported into the FlexNet Manager for Engineering Applications system. From the Contract Summary page, you can delete contracts and sort the contracts by column headings.

Contract Pools

Contract pools are used to define common license terms for a group of product licenses purchased under a contract. These terms can include geographic and other license usage restrictions, and the remix rules. Contract pools have the same license type and the same remix type. A single contract may contain multiple pools, each with different license terms.

Product Licenses

To use licensed software, you purchase product licenses from a software vendor. The contract specifies the terms of the license and indicates how many copies of the product you own. At a particular point in time, the total number of copies is the sum of the quantities of all product licenses that have an effective date on or before the point in time and an expiration date that is either blank—which indicates that the product license is perpetual—or after the point in time.

For reporting and chargeback purposes, a product license can be distributed to different organizational units within your enterprise. For example, if you have a license for 10 copies of Product A, you could distribute six copies to the Broadband division and four copies to the Memory Products division.

In FlexNet Manager for Engineering Applications, you add product licenses to contract pools. After adding the product licenses, you can add license distributions.

License Distributions

You can add a license distribution for a product license, which enables you to capture the number of product licenses that are allocated to your enterprise's various [organizational units](#). For example, if you have a total of 100 licenses for a product, you may want to allocate 30 of them to the Memory Products division and 70 to the Communications division.

License distributions are used primarily for reporting, chargeback, and inventory control purposes; there is no way to restrict product usage.

You can distribute product licenses [automatically](#) or [manually](#). Automatic distribution is based on each organizational unit's percentage usage of the product. Using the previous example, if the Communications division is using the product three times as much as the Memory Products division, they would receive 75 product licenses and Memory Products would receive 25 product licenses.

You can use manual distribution if you know the quantities used by your organizational units—if the quantities are known or fixed and not based on relative usage.

Remixing or Renewing Contract Pools

Remixing is the process of optimizing the money spent on software by adjusting the quantities of product licenses within a set of rules defined by the contract and contract pools. Customers are typically limited to a fixed number of remixes per year over the life of the contract. The remix process uses historical product usage patterns and the limits defined in the contract and pools to forecast the number of licenses your enterprise will require for each of the products purchased under the contract.

A remix is performed on the product licenses within the same contract pool; any manipulation for a remix between the product licenses that belong to different contract pools of a contract are not permitted.

Basic Remix or Renewal Process

The basic process for remixing contract pools using FlexNet Manager for Engineering Applications is as follows. You perform these steps using the FlexNet Manager for Engineering Applications interface. See the FlexNet Manager for Engineering Applications Online Help Library for information specific to each of the steps.

The renewal process is similar to remixing, but the investment decision is related to perpetual licenses and is based on whether your enterprise will choose to renew maintenance.

1. [Add a remix](#) to define the remix parameters, and select the contract pools to be remixed.
2. Using the [View Remix](#) page, specify any additional quantities or buffer percentages for the products in the contract pools being remixed.
3. View the usage trend analysis, as shown by the Product Peak Usage Trend report. (See the *FlexNet Manager for Engineering Applications 2021 R1 Reporting Guide* for information about the available reports.)
4. Override any of the quantities suggested using the [View Remix](#) page.
5. [Complete](#) the remix.

Importing Contract Data

You can import your existing contract data to FlexNet Manager for Engineering Applications using an XML file. The XML schema (contracts.xsd) is located in the FlexNet Manager for Engineering Applications Admin installation directory in the following location: `<fnm_admin_installdir>\site\samples`.

The following is an example of an XML contract import file.

```
<?xml version="1.0" encoding="UTF-8"?>
<vendor name="Mentor Graphics">
  <contract endDate="2011-12-31" fixedCost="1000000" name="Contract One" notes="contract notes"
    paymentAmount="100000" paymentEndDate="2011-10-01" paymentInterval="4" paymentStartDate="2009-04-
    01" startDate="2009-01-01" variableCost="500000">
    <pool annualFeePercent="30.0" discountPercent="60.0" geography="geo notes" initialValue="23456"
      name="Pool One" notes="pool notes" remixablePercent="100.0" remixesPerYear="4"
      shareable="global" site="Site notes">
      <product listPrice="2345" producerName="Mentor Graphics" productName="Advanced Interconnect
      Op SW" productNumber="206143" productVersion="1">
        <license effectiveDate="2009-01-01" notes="License Comments" poNumber="ponum"
        purchaseReq="purch req" quantity="123"/>
        <license effectiveDate="2009-07-01" quantity="22"/>
      </product>
      <product listPrice="44444" producerName="Mentor Graphics" productName="Advanced Rule Set Op
      SW" productNumber="214756" productVersion="1">
        <license effectiveDate="2009-01-01" quantity="333"/>
      </product>
    </pool>
    <pool annualFeePercent="25.0" discountPercent="60.0" forProductionUse="false"
      initialValue="234234" licensingModel="timeBased" name="Pool two" remixablePercent="52.564102"
      remixesPerYear="2" shareable="regional">
      <product listPrice="54444" producerName="Mentor Graphics" productName="Boardsim"
      productNumber="Unknown" productVersion="1">
        <license effectiveDate="2009-01-01" expireDate="2009-12-31" quantity="32"/>
      </product>
      <product listPrice="23423" producerName="Mentor Graphics" productName="Board Station
      Designer" productNumber="39986" productVersion="1">
        <license effectiveDate="2009-01-01" expireDate="2009-12-31" quantity="345"/>
      </product>
    </pool>
  </contract>
</vendor>
```



Task

To import the contract data XML file:

1. From a Command prompt, change to the FlexNet Manager for Engineering Applications Admin installation directory.
2. Type the following command:

```
flexnet importcontracts -u<user> -p<password> -f<filename_with_path> [-s<server>]
```

Software Vendors

Software vendors are the organizations that sell licensed software to your enterprise. The terms of the license are defined in a contract.

The software vendor may be the same organization as the software producer.



Note • *LinkRight Tracker used the term vendor to refer to both the company that produces software and the company that sells software. In FlexNet Manager for Engineering Applications, these two concepts are represented by software producers and software vendors, respectively. The software producer and the software vendor may be the same company.*

Product Choices

Product choices are automatically created by the product usage aggregation process. FlexNet Manager for Engineering Applications uses a data aggregation process (also referred to as a roll-up) to determine which products are being used at your enterprise, based on the feature usage data generated by report logs. FlexNet Manager for Engineering Applications makes the product determinations based on the feature-to-product associations that you provided.

The Product Choices page displays a list of all product choices, including the following information for each:

- **Possible Products**—List of the products that the product usage aggregation process had to choose from when determining product usage. It is a list of products that share the features listed in the Common Features column.
- **Common Features**—Features that each of the possible products have in common.
- **Chosen Product**—Name of the software product that the product usage aggregation process chose for the product usage.
- **Product Choice Override**—Name of the software product that the user chose to override the product chosen by the product usage aggregation process.

Product Aggregation

Product aggregation refers to the portions of the aggregation process that are responsible for populating the product-centric areas of the data warehouse. This allows for reporting to be performed on a per-product basis, rather than focusing on simple feature usage. In addition to reporting, product aggregation is also necessary to enable the Investment Planning functionality, including remixing.

If you have purchased Investment Planning functionality, product aggregation is automatically performed during every aggregation process. During aggregation, feature events (imported from report logs) are considered chronologically, and contribute to the usage summaries for individuals and organizations, for specific hours. These usage summaries are stored in the measure tables of the data warehouse for reporting. Product aggregation is similar, but includes two additional phases: the creation of *product events* from feature events, and *product resolution*.

Product Event Creation

Each product is defined (either using the Products page in the FlexNet Manager for Engineering Applications interface or by importing a product-definition XML file) to include a specific set of features. In order to generate product usage summary data similar to the feature usage summary data produced in aggregation, FlexNet Manager for Engineering Applications must first create product events from the feature events being processed.

As feature events are processed chronologically, each individual feature event is compared to the product definitions. If the feature in use is found to be part of one or more products, it is either added to an existing product event, or a new product event is created.

When a new product event is created, it is assigned the start and end times that correspond to the start and end time of the feature event currently being processed, and a list of possible products is assigned. This set of possible products is the set of all products that include the feature in question.

When a feature event is added to an existing product event, the end time of that product event is extended, if necessary, to the end time of the feature event. Also, the list of possible products is reduced to only contain products that include all the features in the product event.

The decision to add to an existing product event is based on the following questions:

- Is the start time of this feature event before the end time of the product event?
- Is this feature included in one or more of the possible products for that product event?
- Is the feature available (not already in use) in this product event at the feature event's start time?

If the answer to all of these questions is yes, the feature event is added; otherwise, a new product event is created.

Because feature events are processed chronologically, once the aggregation process reaches a feature event that starts after the end time of a specific product event, it is safe to say that no more feature events will be added to that product event.

Product Resolution

When FlexNet Manager for Engineering Applications is finished adding feature events to a product event, it must determine what product this event is for. In the simplest case, there would only be one possible product that included all the features that contributed feature events to the product event, and there is then no choice that needs to be made. However, if there is more than one possible product—due to multiple products including the same features—then the aggregation must resolve this list of possible products to a single product.

The product resolution is dictated by four rules of comparison, which are considered in the order listed:

1. The product with the lowest net cost is selected. If this still results in more than one possible product, then the next rule is evaluated.
2. From products identified by rule 1, the product with the least number of features is selected. If this still results in more than one possible product, then the next rule is evaluated.
3. From products identified by rule 2, the product with the largest number of licenses is selected. If this still results in more than one possible product, then the next rule is evaluated.
4. The first product created (with the lowest ID in the database) is selected.

Before these comparison rules are applied, the aggregation will first check to see that all products have valid licenses. If some do, and others do not, then only the products with valid licenses are considered. However, if none of the possible products have valid licenses, then all products will be considered; however, without product licenses, only rules 2 and 4 will have any affect on the product-resolution process.

When these rules are applied, product choice objects are created, and these product choices are displayed on the Product Choices page in the FlexNet Manager for Engineering Applications interface. There, you can override the product choice with another product, and the override you enter will be applied in future product aggregations.

Glossary

The following table defines many of the terms that are used throughout the FlexNet Manager for Engineering Applications documentation.

Term	Definition
concurrent license	Entitlement defined as a fixed number of licenses available to be shared among a group of users. Also referred to as network or floating licenses.
contract	<p>A document that defines the terms under which an enterprise purchases product licenses from a software vendor. These terms typically include the contract's start and end dates, the fixed and variable costs, remix rules for the contract, and any license usage restrictions (for example, geographic restrictions).</p> <p>Product licenses purchased under a contract are grouped into one or more license pools.</p>
counted license	A license with a non-zero license count. A license server is required to manage a counted license. Counted licenses can either float on a network or be node-locked.
data aggregation	The process through which FlexNet Manager for Engineering Applications determines which products are being used, based on the feature usage contained in report logs.
daemon	A process that “serves” clients. Sometimes referred to as a server or service process.
debug log file	One or more ASCII text files written by the license server. A debug log file contains status and error messages useful for debugging the license server.
duplicate grouping	Duplicate license requests from the same user, vendor, host or display share one license.
end user	Your customer to whom the FlexEnabled product is supplied.

Term	Definition
end-user license administrator	The staff member or members at the end-user site chartered with installing the FlexEnabled product and administering the license file and server as appropriate. This could be the same person as the actual user of the product.
entitlement	The characteristics that define the legal right to use the feature or product.
feature	<p>A separately licensed unit of functionality in a product. The meaning of a feature depends entirely on how it is used by a product developer. For example, a feature could represent any of the following:</p> <ul style="list-style-type: none"> ● A product consisting of one or more components ● A specific version of a product ● A part of a product ● A piece of data (restricted via the access routines) <p>One feature can be included as a part of many different products.</p>
feature license	<p>A license to use a discrete unit of functionality in a software product. A software vendor delivers feature licenses to the enterprise in one or more license files.</p> <p>Feature licenses are distributed over a network by a vendor daemon running on a license server. Feature licenses are used when an end user accesses licensed parts of a software product.</p>
feature license event	A feature license event records the usage of a feature license by an end user. The license server creates feature license events each time a feature license is used.
feature line	A line in a license file that licenses a particular feature. A feature line begins with one of the keywords FEATURE, INCREMENT, or UPGRADE.
FlexNet Agent	An application deployed on each FlexNet license server machine that provides communication between one or more license servers and FlexNet Manager for Engineering Applications.
FlexNet Manager for Engineering Applications	An application deployed at an enterprise to manage FlexNet licenses and license servers. FlexNet Manager for Engineering Applications comprises two modules—FlexNet Manager for Engineering Applications Admin and FlexNet Manager for Engineering Applications Reporting.
FlexNet Publisher components	The individual parts the vendor delivers to the enterprise that compose the FlexNet Publisher solution.
FlexNet Publisher toolkit	Contains all the necessary components and libraries used by the software publisher to create a FlexEnabled product.

Term	Definition
FlexEnabled product	A product, instrumented with calls to the FlexNet Publisher client library, that requests a license.
floating license	A license that can authorize usage of a product by one of a group of users on a network. A license server is required to manage a floating license. Floating licenses can either be counted or uncounted. An uncounted floating license is effectively a site license.
fulfillment record	A data structure specific to a publisher's FlexEnabled application that contains license rights.
heartbeats	Periodic messages sent from a FlexEnabled product to a license server to solicit replies that ensure that the license server is still running. Heartbeats can be implemented automatically or called explicitly from a FlexEnabled product. If called explicitly, they are referred to as manual heartbeats.
hostid	A means used to uniquely identify a specific system.
hostid type	The scheme in which the hostid is determined for a particular system architecture.
license	The legal right to use a feature. FlexNet Publisher can restrict licenses for features by counting the number of licenses already in use for a feature when new requests are made by the FlexNet Publisher product. FlexNet Publisher can also restrict product usage to particular node or user name.
license certificate	One or more feature lines specific to a vendor's FlexEnabled product. This certificate is provided to the end user who incorporates it into a license file.
license file	<p>A license file is a collection of feature licenses. It is a text file specific to an enterprise site containing the license certificates from one or more software publishers and/or vendors. The enterprise installs the license file on a license server.</p> <p>This file can contain descriptions of the following, for one or more FlexEnabled products:</p> <ul style="list-style-type: none"> ● License server node or nodes ● Vendor daemon or daemons ● Licenses
license-file list	A list of license files separated with a colon ":" on UNIX and a semi-colon ";" on Windows. A license-file list can be accepted in most places where a license file is appropriate. When a directory is specified, all files matching *.lic in that directory are automatically used, as if specified as a list.
license key	A 12- to 20-character hexadecimal license signature used to authenticate a license. See also signature .

Term	Definition
license model	The set of defining characteristics for product entitlement. The fundamental layer of a licensing policy.
license pool	License pools are used to define common license terms for a group of product licenses purchased under a contract . These terms include remix rules and license usage restrictions.
license server	A license manager daemon (lmadmin or lmgrd) and one or more vendor daemon processes. License server refers to the processes, not the computer on which they run.
license server machine	A computer system that runs the license server processes. The license server machine hosts all site-specific information regarding licensed feature usage. Multiple license server machines used for three-server redundancy can logically be considered the license server machine. Also known as license server node.
license sharing	Duplicate license requests from the same user, host or display share one license. See duplicate grouping .
license system	A system used to license product features. FlexNet and IBM's LUM are examples of license systems.
licensing policy	The constraints applied to the legal right to use a feature.
lmgrd	The daemon that sends client processes to the correct vendor daemon on the correct machine. The same license manager daemon process can be used by any application from any vendor because this daemon neither authenticates nor dispenses licenses. lmgrd processes no user requests on its own, but forwards these requests to a vendor daemon.
network license	Entitlement defined as a fixed number of licenses available to be shared among a group of users across a network. Also referred to as concurrent or floating licenses.
node-locked license	A license that can authorize use of a product to be run on a specific machine based on a unique identifier known as a hostid. Node-locked licenses can be counted or uncounted. Node-locked, uncounted licenses do not require a license server.
options file	A text file implemented for a particular vendor's product by a FlexNet license administrator at an end-user site to customize the behavior of the vendor daemon or to enable or restrict the use of the licenses managed by the vendor daemon at that site.
package	A set of independent components to be licensed as a single product. Packages are defined in the license certificate, not in the product; this allows a vendor to change the components of a package until just prior to shipping the license.
package suite	A set of products combined into a package with the restriction that the package components of a single license may not be simultaneously shared.

Term	Definition
product	Software that comprises one or more licensed features .
product choice	A product choice captures the results of the product resolution process for each unique set of possible products to choose from.
product license	Defined in a contract , the product license specifies the quantity purchased, pricing information, and the license start and expiration dates.
product license event	A product license event records the usage of a product license by a user. Product license events are not created directly by the license server; they are created by the product usage aggregation process by combining feature license events.
product resolution	The process of determining which product was used when the feature license events could have come from more than one product.
product usage aggregation	The process of creating product license events by analyzing feature license events. This is how FlexNet Manager for Engineering Applications identifies which products were used.
remix	Remixing is the process of optimizing the money spent on software by adjusting the quantities of Product Licenses within a set of rules defined by the Contract and Pools. Customers are usually limited to a fixed number of remixes per year over the life of the contract. The remix process uses product usage patterns and the limits defined in the contract and pools to forecast the number of licenses required for each of the products purchased under the contract.
report log file	<p>The feature usage data written by a single vendor daemon. Report logs are not human readable; the data is compressed and authenticated for use with FlexNet Manager for Engineering Applications.</p> <p>FlexNet Manager for Engineering Applications uses the data contained in the report log files to produce various usage reports.</p>
rotation	The action of switching the report log file by moving the existing report log information to a new file, then starting a new report log with the original report log file name.
served license	A license that requires the management of a license server.
signature	A secure, 12- to 120-character hexadecimal number that authenticates the readable feature line in the license file, ensuring that the license text has not been modified. See also license key .
software producer	A company that makes software products that use a licensing system to control access to various functions (features) of the product. The software producer may also be the software vendor.
software vendor	A company that sells licensed software products. The software vendor may also be the software producer.

Term	Definition
suite	See package suite .
supersede	A way of changing characteristics of a license with new ones, thereby making obsolete the earlier ones. For example, a supersede license can be issued to both change the name of a feature and obsolete the older name at the same time. This is more powerful than upgrading.
tamper-resistant licenses	An encryption method employing an industry-recognized asymmetric, public/private key digital signature.
unassigned feature	A feature that is not associated with any products .
uncounted license	A license that does not restrict the number of uses of a feature. Usually node-locked to a machine hostid.
unserved license	A license that does not require the management of a license server.
upgrade	The concept of upgrading the version entitlement for a given license certificate.
vendor	See software vendor .
vendor alias	The alias given to one or more vendor daemon names that symbolically represents the vendor's name from whom the daemons are deployed.
vendor daemon	The server process that dispenses licenses for a particular software publisher's features. The software publisher builds this binary using libraries supplied by Flexera Software.
vendor name	A unique vendor name that is supplied to the software publisher by Flexera Software. This is used as the name of the vendor daemon.

Index

A

- activatable license [23](#)
- activation
 - debug logging [18](#)
 - report logging [18](#)
- adding
 - products [38](#)
- additional licenses [34](#)
- administrator, end-user [48](#)
- agent [8](#), [48](#)
- aggregation [47](#), [51](#)
- analyzing your data, strategies for [27](#)
- anchor [22](#)
- architecture, FlexNet Manager for Engineering Applications [7](#)
- assigning features to products [40](#)
- audience of manual [5](#)
- Auto Restart [18](#)

B

- balancing license load [34](#)
- basic usage concepts [28](#)
- binding [22](#)

C

- common vendor daemon
 - mapping [19](#)
- components, FlexNet licensing [48](#)
- concurrent license [22](#)
- concurrent licenses [47](#)
- configuring the license server [17](#)
- contacts
 - adding for software producers [40](#)

- contract [47](#)
- contract pools [41](#)
- contracts [41](#)
 - importing [43](#)
 - remixing [42](#)
- counted license [47](#)

D

- daemon [47](#)
- daemon, vendor [52](#)
- data aggregation [47](#)
- data, strategies for analyzing [27](#)
- debug log
 - definition [47](#)
- debug log location [18](#)
 - vendor daemon [18](#)
- debug logging activation [18](#)
- denials
 - eliminating spurious [30](#)
 - in raw reports [28](#)
 - license [28](#)
 - reasons [28](#)
 - ultimate [30](#)
- deployment [8](#)
- detachable license [22](#)
- determining flexnet license server status [17](#)
- display name, license server [18](#)
- distributions
 - adding [42](#)
- documentation, additional [6](#)
- duplicate grouping [47](#)

E

eliminating spurious denials [30](#)
 end user [47](#)
 end-user license administrator [48](#)
 entitlement [48](#)
 exporting product definitions [40](#)

F

feature [48](#)
 unassigned [52](#)
 feature license [48](#)
 feature license event [48](#)
 feature line [48](#)
 features [40](#)
 assigning to products [40](#)
 features, remixing licenses within [35](#)
 file
 debug log [47](#)
 license [49](#)
 options [50](#)
 report log [51](#)
 FlexEnabled product [49](#)
 Flexera Analytics [8](#)
 FlexNet [48](#)
 FlexNet Agent [8](#), [48](#)
 FlexNet licensing components [48](#)
 FlexNet Manager for Engineering Applications
 architecture [7](#)
 getting started [9](#)
 user interface [10](#)
 FlexNet Publisher licensing toolkit [48](#)
 floating license [49](#)
 fulfillment record [22](#), [49](#)

G

getting started [9](#)
 glossary [47](#)
 group name, vendor daemon [18](#)

H

heartbeats [49](#)
 host name, license server [18](#)
 hostid [49](#)
 hostid type [49](#)

I

importing contract data [43](#)
 importing product definitions [38](#)

interface
 license server [11](#)
 interval
 report log rotation [18](#)
 investment planning [37](#)

K

key, license [49](#)

L

license [49](#)
 counted [47](#)
 feature [48](#)
 floating [49](#)
 node-locked [50](#)
 product [51](#)
 served [51](#)
 sharing [50](#)
 uncounted [52](#)
 unserved [52](#)
 license certificate [21](#), [49](#)
 definition [49](#)
 license counts [23](#), [24](#)
 license denials [28](#)
 license distributions
 adding [42](#)
 license events
 feature [48](#)
 product [51](#)
 license file [21](#), [49](#)
 license groups [22](#)
 license key [49](#)
 license load, balancing [34](#)
 license manager [18](#)
 debug log location [18](#)
 options [18](#)
 port [18](#)
 status [17](#)
 license manager daemon [50](#)
 license model [50](#)
 license pool [50](#)
 license pools [23](#)
 license rights representation [21](#)
 license server [50](#)
 configuring [17](#)
 display name [18](#)
 host name [18](#)
 starting [13](#), [15](#)
 status, determining [17](#)
 stopping [14](#)
 user interface [11](#)
 license server machine [50](#)

- license system 50
- license usage, overdraft 24, 32
- license-file list 49
- licenses
 - product 41
- licenses, adding 34
- licensing policy 50
- line, feature 48
- lmgrd 50
- load balancing, license 34
- location
 - license manager debug log 18
 - vendor daemon debug log 18
- log file
 - debug 47
 - report 51
- logging activation
 - debug 18
 - report 18

M

- machine, license server 50
- mapping
 - common vendor daemon 19
- mapping features to products 40
- model, license 50

N

- name, report log 18
- name, vendor 52
- name, vendor daemon group 18
- network license 50
- node-locked license 50

O

- options
 - license manager 18
 - vendor daemon 18
- options file
 - definition 50
- overdraft license usage 24, 32

P

- package 50
- package suite 50
- Planner tab 37
- policy, licensing 50
- pool
 - license 50

- pools
 - contract 41
- port, license manager 18
- preface 5
- prerequisites 9
- producer 51
- product 51
- product choice 51
- product choices 44
- product definition XML 39
- product license 51
- product license event 51
- product licenses 41
- product resolution 51
- product usage aggregation 51
- products
 - adding 38
 - assigning features 40
 - exporting definitions 40
 - importing definitions 38
- products.xsd 39
- ProductSample.xml 39

R

- raw report
 - denials in 28
- refresh rate 18
- remix 51
- remixing 42
- remixing licenses 35
- report
 - denials in raw 28
 - raw text 28
- report log
 - activation 18
 - name 18
 - rotation interval 18
- report log file 51
- rotation
 - report log 51
- rotation interval, report log 18

S

- sample files
 - product definition XML 39
- served license 51
- server
 - configuring 17
 - starting 13, 15
 - stopping 14
- server display name 18
- server host name 18

- server machine, license [50](#)
- server status, determining [17](#)
- server user interface [11](#)
- server, license [50](#)
- signature [51](#)
- software producer [51](#)
- software producers [40](#)
 - adding contact information [40](#)
- software vendor [51](#)
- software vendors [44](#)
- spurious denials, eliminating [30](#)
- starting
 - license server [13](#), [15](#)
 - vendor daemon [16](#)
- status
 - determining flexnet license server [17](#)
 - license manager [17](#)
 - vendor daemon [17](#)
- stopping
 - license server [14](#)
 - vendor daemon [16](#)
- strategies for analyzing your data [27](#)
- suite, package [50](#)
- supersede [52](#)

T

- tamper resistant licenses [52](#)
- Trusted ID [22](#)
- trusted storage [22](#)
- type, hostid [49](#)
- typographic conventions [6](#)

U

- ultimate denials [30](#)
- unassigned feature [52](#)
- uncounted license [52](#)
- unserved license [52](#)
- upgrade [52](#)
- usage concepts [28](#)
- user interface [10](#)
 - license server [11](#)
- users, notifying [15](#), [16](#)

V

- vendor [51](#)
- vendor alias [52](#)
- vendor daemon [52](#)
 - debug log location [18](#)
 - group name [18](#)
 - option [18](#)

- starting [16](#)
- status [17](#)
- stopping [16](#)
- vendor daemons
 - associating with software producers [41](#)
 - common vendor daemon mapping [19](#)
- vendor name [52](#)

X

- XML
 - for importing contract data [43](#)
 - product definition format [39](#)