



FlexNet Manager for Engineering Applications 2022 R2

Reporting Guide

Legal Information

Book Name: FlexNet Manager for Engineering Applications 2022 R2 Reporting Guide
Part Number: FNMEA-2022R2-RG01
Product Release Date: December 2022

Copyright Notice

Copyright © 2022 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 is strictly prohibited. Except where expressly provided by Flexera in writing, possession of this technology shall not be construed to confer any license or rights under any Flexera intellectual property rights, whether by estoppel, implication, or otherwise.

All copies of the technology and related information, if allowed by Flexera, 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 download site.

Intellectual Property

For a list of trademarks and patents that are owned by Flexera, see <https://www.flexera.com/company/legal/intellectual-property.html>. All other brand and product names mentioned in Flexera 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 | Overview | 7 |
| | Reporting Models | 7 |
| | Reporting Workflow | 8 |
| 2 | Classic Reporting | 11 |
| | Reporting Tab | 11 |
| | Generating and Viewing Classic Reports | 12 |
| | Creating Report Configurations | 12 |
| | Generating a Report Based on a Report Configuration | 12 |
| | Report Output Formats | 13 |
| | Output in Report Interchange Format | 14 |
| | Working with FlexNet Publisher Report Logs | 15 |
| | Enabling Report Logging | 15 |
| | Managing (Rotating) Report Logs | 16 |
| | Report Log Administration | 17 |
| | Sample Report Log Data | 17 |
| | Report Log Timestamps and Time Zones | 18 |
| | Specifying Post-Processing Options | 18 |
| | Report Output Directory | 19 |
| | Output Directory Options | 19 |
| | Post-Processing Command | 19 |
| | Bulk Importing Existing Usage Data | 26 |
| | Report Types | 28 |
| | Usage Over Time Report | 28 |
| | High Water Mark Report | 29 |
| | Peak Usage and Denial Report | 30 |
| | Summary Barchart Report | 31 |
| | Usage Efficiency Report | 34 |

| | |
|---|-----------|
| Usage Summary Report | 35 |
| Unique User Report | 39 |
| Server Coverage Report | 40 |
| Raw Report | 41 |
| 3 Flexera Analytics Reporting | 43 |
| Components | 43 |
| Databases and Servers | 43 |
| Reporting Tools | 44 |
| Flexera Analytics Reports | 44 |
| Data Warehouse and Aggregation | 44 |
| Denial Folding | 44 |
| Aggregation Configuration Settings | 45 |
| 4 Organizational Structures for Reporting | 47 |
| Defining Organizational Structures | 47 |
| Defining Organizational Structures Using an XML File | 48 |
| Sample XML Organizational Structures | 52 |
| Importing Organizational Structure XML Files | 55 |
| Reporting on Usage By Project | 55 |
| Defining Organizational Structures Using the FlexNet Manager for Engineering Applications Interface | 56 |
| 5 Usage Data Maintenance | 57 |
| Report Log Usage Data Maintenance | 57 |
| Aggregation Usage Data Maintenance | 58 |
| Best Practices for Usage Data Maintenance | 58 |
| 6 Reporting on non-FlexNet Application Usage | 61 |
| Reporting for IBM LUM | 61 |
| Process Overview | 61 |
| Configuring LUM License Servers to Log Events | 62 |
| Processing LUM Report Logs Using i4blt | 62 |
| Splitting LUM Files Using processLUMFile Utility | 62 |
| Reporting on non-FlexNet and non-LUM Application Usage | 63 |
| 7 Classic Reporting Data Model | 65 |
| RPT_CATALOG_SECTION Table | 67 |
| RPT_FEATURE Table | 67 |
| RPT_VENDOR_DAEMON Table | 70 |
| RPT_ERROR Table | 71 |
| RPT_REASON Table | 73 |
| RPT_USAGE_LIC_EVENT_153 Table | 73 |
| RPT_USAGE_NOLIC_EVENT_153 Table | 77 |

| | |
|---|------------|
| RPT_USAGE_LIC_DEVENT_153 Table | 79 |
| 8 Generating Command-Line Reports | 83 |
| Running Reports from the Command Line | 84 |
| Defining Input and Output Files..... | 85 |
| Defining the Level of Detail in a Report | 86 |
| Defining Field Filters | 87 |
| Customizing Appearance of Graphical Report Output | 89 |
| Using a Configuration File..... | 91 |
| Specifying Feature Names in Configuration Files | 91 |
| Defining a Raw Report..... | 91 |
| Command-Line Reference | 92 |
| Reporting Command-Line Parameters..... | 92 |
| Breakdown Categories..... | 97 |
| Filter Categories..... | 98 |
| Raw Report Fields | 100 |
| 9 Index | 103 |

1

Overview

FlexNet Manager for Engineering Applications provides a comprehensive reporting service, providing both traditional (Classic) reporting of historical usage data, and additional features using Flexera Analytics (powered by Cognos) to enable faster reporting of your enterprise's usage data. Flexera Analytics also enables you to design your own ad-hoc reports for your specific business needs.



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.*

FlexNet Manager for Engineering Applications Reporting provides the following features:

- Flexible reporting presentation—including charts, tables, and text-based reports.
- Simple and complex report-authoring tools, to support both novice and advanced users.
- The ability to schedule and distribute reports in common formats like PDF, HTML and CSV.

You should be familiar with FlexNet Manager for Engineering Applications features before running reports on your usage data. See the *FlexNet Manager for Engineering Applications Concepts Guide* for an introduction to FlexNet Manager for Engineering Applications, and the *FlexNet Manager for Engineering Applications Help Library* for additional information about reporting on usage data.

Reporting Models

FlexNet Manager for Engineering Applications Reporting supports two reporting models:

- **Classic Reporting** (traditional FlexNet Manager for Engineering Applications reports)—FlexNet Manager for Engineering Applications Classic reporting generates reports using the Transactional Reporting data model. Classic reporting provides a variety of report configurations that you can modify, but it requires more data processing per report. If you need to generate a report based on a considerable amount of data, you can use FlexNet Manager for Engineering Applications's report scheduler, which enables you to schedule the report to run at a specified time (for example, overnight) and view the report later. Classic reports are static and are not interactive. See [Chapter 2, "Classic Reporting."](#)

- Flexera Analytics (Cognos reporting)—FlexNet Manager for Engineering Applications includes access to Cognos reporting functionality, which provides quick, online review of historical data. Cognos uses a different structure to model data than that used for classic reporting, and can deliver reports more quickly. Many Flexera Analytics reports offer drill-down capability, which enables you to click through to see more detailed data for a piece of report data. Flexera Analytics also provides custom reporting capability, which enables you to design your own specialized reports.



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

For more information, see [Chapter 3, Flexera Analytics Reporting](#). Cognos documentation for both Query Studio and Report Studio is available from the Cognos Web site at www.cognos.com.

Reporting Workflow

Figure 1-1 shows how reports are produced in FlexNet Manager for Engineering Applications.

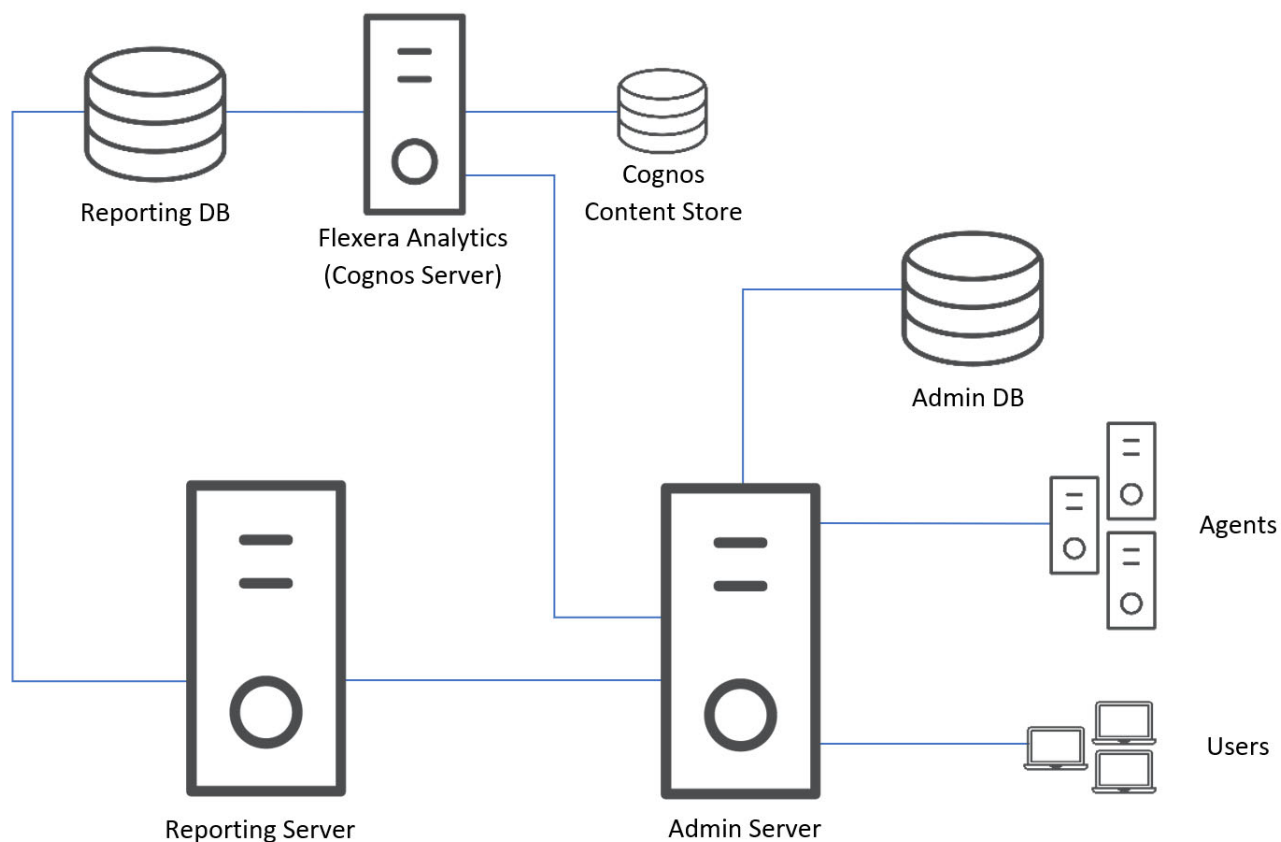


Figure 1-1: FlexNet Manager for Engineering Applications reporting workflow

After report log rotation, manual report log import or bulk import, only the classic reports are available for the imported usage data. Flexera Analytics reports for the data are available only after data aggregation is complete. Data aggregation is the process by which FlexNet Manager for Engineering Applications takes the report log data and uses it to populate the data warehouse, which is the set of database tables on which Flexera Analytics reports are based.

Classic Reporting

FlexNet Manager for Engineering Applications Classic Reporting is a valuable tool to aid you in analyzing license usage patterns. When you know how licenses are being used within your enterprise, you can make informed license management decisions about more licenses or the load requirements of your license servers.

FlexNet Manager for Engineering Applications generates reports based on FlexNet Publisher license-usage activity from usage report logs on a per-vendor daemon basis. See [Working with FlexNet Publisher Report Logs](#) for more information on report logs.

The basic steps involved in generating a report include:

1. Enabling FlexNet Publisher report logs. See [Working with FlexNet Publisher Report Logs](#).
2. Deciding on the type of report. See [Report Types](#).
3. Creating a report configuration or using a previously created configuration.
4. Generating the report and viewing the result.

See Chapter 5, “Analyzing FlexNet License Usage”, in the *15.14 Concepts Guide*, for ways in which you can analyze usage data and determine your licensing needs.

You can also use command-line interface reporting to design your own reports. See [Appendix 8, Generating Command-Line Reports](#) for more information.

Reporting Tab

FlexNet Manager for Engineering Applications provides several predefined report types that enable you to quickly generate reports. The reports can summarize usage data (Summary reports), show individual usage events (Raw reports), or document the time that a license server is running or up (Server reports). For most report types, you can specify the features, time periods, users, hosts, IP addresses, displays, vendor-defined strings, and event types (used, denied, and so on) to include in the report. See the [Report Types](#) section for more information.

You access FlexNet Manager for Engineering Applications Classic reporting functionality by clicking the Reporting tab on the user interface, and then clicking the Classic tab. This displays the Report Configurations page, where you can edit, schedule, and run predefined Classic reports.

You can save customized report configurations. In addition, you can save combinations of report parameters in configuration files that you retrieve later in order to generate a given report on demand. For information about customized report configuration, see the FlexNet Manager for Engineering Applications Online Help Library.

Generating and Viewing Classic Reports

All predefined Classic report configurations are listed on the Report Configurations screen, available from the Reports tab.

Creating Report Configurations

Before you can generate a report, you need to create a report configuration. The report configuration defines the parameters used in creating the reports from the set of available report logs. To make report logs available, either configure FlexNet Manager for Engineering Applications to collect report logs from your license servers (see [Managing \(Rotating\) Report Logs](#)) or manually import them.

Report configurations are created using the Report Configurations page. You can edit an existing configuration or create a new one based on a predefined report type. In either case, FlexNet Manager for Engineering Applications presents a series of screens that walk you through the report-configuration process. (See the FlexNet Manager for Engineering Applications Online Help Library for information on configuring a particular report.)

You can also copy an existing report configuration. This enables you to easily create a report that has the same report specifications with a few exceptions (for example, a different date range, vendor daemon, or host server).

Generating a Report Based on a Report Configuration



Task

To generate a report based on a configuration:

1. Click the **Run** button for the report configuration entry that you want.
2. Click the **Report Inbox** subtab and monitor the status of the generation on the **Report Inbox** page. (You may need to refresh the page.)
3. When the status is Complete, you can view the report by:
 - Clicking the Text or Image link under the Configuration Name
This displays the report in a new browser window, keeping the FlexNet Manager for Engineering Applications interface available in the original browser window.
 - Clicking the **Download** button for that report
This archives all report components into a .zip file and displays a folder of file names using a ZIP utility. From here, use the ZIP utility according to standard practice. Use the browser **Back** button to return to the FlexNet Manager for Engineering Applications user interface.
 - Checking the report output directory
As part of creating a report configuration, you can specify an output directory of your choice that FlexNet Manager for Engineering Applications writes reports to.



Note • If you are running FlexNet Manager for Engineering Applications using a demo license (node-locked to hostid DEMO), you are limited to generating reports whose report log data is dated within 45 days of the license expiration date. For information about how FlexNet Manager for Engineering Applications is licensed, see the FlexNet Manager for Engineering Applications 2020 R1 Installation Guide.

Report Output Formats

FlexNet Manager for Engineering Applications supports several report formats (Table 2-1): text, PNG, JPEG, HTML, XML and Report Interchange Format (RIF).

- Graphical report output is saved to files in PNG or JPEG format.
- HTML report output can be viewed in an HTML browser.
- RIF can be used to export report data to another software program, such as a spreadsheet. See [Output in Report Interchange Format](#) for more information.

Table 2-1 • Report Output Formats

| Report Family | Report Type | Output Options | Output Formats |
|----------------|------------------------|----------------|------------------------|
| Summary | Usage Over Time | Image Text | PNG or JPEG RIF |
| | Peak Usage and Denials | Image Text | PNG or JPEG |
| | High Water Mark | Image, Text | PNG or JPEG RIF |
| | Summary Barchart | Image | PNG or JPEG |
| | Unique User Report | Text | HTML or XML |
| | Usage Efficiency | Image, Text | PNG or JPEG RIF |
| | Usage Summary | Text | Text, HTML, RIF |
| Server | Server Coverage | Text | Text |
| Raw | Raw | Text | Text, RIF, XML |

Output in Report Interchange Format

Output in Report Interchange Format (RIF) is available for all reports, except Peak Usage and Denial, Summary Barchart and Server Coverage reports. RIF is a simple, flexible, plain-text format that is designed to pass tabular information between the reporting module and another application, such as a spreadsheet.

The default field separator is a space character, although you can specify a different delimiter in the Output Options step of the Create Report Configuration wizard. Fields that contain the specified field delimiter are surrounded in quotation marks.

Textual reports are written to a single RIF file; each page of a multi-page graphical report is written to a separate RIF file. Sequential numbers (starting with 0) are appended to the filename base of the RIF files and they have the extension `.rif`. A Table of Contents RIF file is generated for each RIF report and is named `output_filename_base_toc.rif`. In addition, a Usage Efficiency report generated in RIF format has “_e” at the end of the file name base, for example, `output_filename_base_e.rif`.

Report Interface Format Specification

```
#BEGIN HEADER
header_Lines
#END HEADER
#cols [#rows]           // These lines contain
column_Labels           // info for the first
[rows]                  // table.
```



Note • When parsing RIF files, it is best to select columns by name so that, as new columns are added to the report, previously developed programs that read the RIF file are not affected.

More than one table may be contained in a RIF file. The `#rows` entry is not present if the file contains rows from only one table. If the `#rows` entry is missing, read all of the rows until the end of the file. If there is more than one table in the RIF file, the table-specific portion of the format is repeated for each additional table.

```
#cols #rows             // These lines contain
column_Labels           // an additional
[rows]                  // table.
```

If you break down data by more than one category in a Usage Summary report written in RIF, each of the categories is presented in its own column. RIF output from a Usage Summary report does not contain a column for the total number of license requests—the total can be obtained by adding the used + denied + unsupported license request columns in the RIF output.

Overdraft and unlicensed usage is shown in the Type column of a Usage Efficiency report written in RIF. The Type column appears in RIF output only if there are overdraft licenses available or unlicensed usage in the report log. Overdraft license usage is designated OVERDRAFT in the Type column. If the number of licenses used exceeds the number of licenses available, either because of corruption of the report log or by licenses in use during a reread, the designation is NO_LICENSE. Usage of non-overdraft licenses is designated REGULAR.

Working with FlexNet Publisher Report Logs

Report log files contain feature usage information and are generated by vendor daemons. However, a vendor daemon does not write report logs by default; this action must be enabled. The data in report logs is compressed and encrypted and is used only by FlexNet Manager for Engineering Applications reporting.

Enabling Report Logging

Follow the instructions in this section to enable vendor daemon report logging. Each report log that FlexNet Manager for Engineering Applications processes must contain usage data written by a single vendor daemon running on a single machine.



Task

To enable report logging, for each instance of a vendor daemon:

1. Create an options file (or edit the existing options file) to include the following line—

```
REPORTLOG +path_to_report_Log
```

—where *path_to_report_Log* is the full path name to the report log, and the filename component of *path_to_report_Log* is of the form *vendor.r1*.

The leading “+” character tells the vendor daemon to append to that file each time it starts, rather than clearing and overwriting the file. There is no space after the “+” character.



Note • Refer to the *License Administration Guide* (part of the FlexNet Publisher documentation set) for information about creating and editing options files. The *License Administration Guide* is available from the Flexera Software download site, *FlexNet Operations On Demand* (<https://flexerasoftware.subscribenet.com>).

2. Save the options file in the form *vendor.opt*, where *vendor* is the name of the vendor daemon.
3. Add the options file path to the VENDOR line in the license file. For example:

```
VENDOR vendor options=options_file_path
```

Surround *options_file_path* with double quotes if it contains spaces.

4. Direct the vendor daemon to reread the license file.



Note • When you change or create an options file for a vendor daemon later than version 8.0, you must restart the vendor daemon in order for those changes to take effect.

Example: Enabling Report Logging

To make vendor daemon *myvendor* write a report log to */usr/app/logs/myvendor.r1*, do the following:

1. Create an options file named */usr/app/license/myvendor.opt* containing one line:

```
REPORTLOG +/usr/app/logs/myvendor.r1
```

2. Edit the license file, adding the options file path to the VENDOR line. The VENDOR line might look like this:

```
VENDOR myvendor options=/usr/app/license/myvendor.opt
```
3. In the FlexNet Manager for Engineering Applications interface, click the Servers tab, then click Vendor Daemons, and select the display name for myvendor.
4. Direct the vendor daemon to reread the license file by clicking the Reread License Files button on the Configure and Manage Vendor Daemons page.

Redundant License Server Configuration

The options file and license files for each license server in a redundant license server configuration must be set up in order for report logging to continue when the master license server changes. If a master license server fails, the new master reads its license and options files and writes to its own report log. Additionally, each vendor daemon in the configuration must write to a separate report log.

Managing (Rotating) Report Logs

As a vendor daemon runs for a longer period of time, the size of the report log increases. If your enterprise has a lot of license usage activity, the report log can become very large in a short period of time.

FlexNet Manager for Engineering Applications provides the capability to automatically archive the usage data from the active report log to a new archive report log file, thus flushing out the active report log file to zero bytes. This process is known as *rotating* the report log. See “[Report Log Administration](#),” for details about archived report logs.

Scheduling Report Log Rotation

You can configure a rotation schedule based on the rate at which usage data is generated. Each time the active report log is rotated, the resulting new archive file is cataloged and becomes available to report configurations.

When you schedule the report log rotation time, the time you need to specify is based on the FlexNet Manager for Engineering Applications server time, and not your local time, if those times differ.



Task

To set up an automatic report log rotation schedule:

1. From the Servers tab, click Vendor Daemons.
2. Click the vendor daemon’s display name.
3. Modify parameters in the Report Log section of the Configure and Manage Vendor Daemon page.
4. Click Save.

Manually Rotating Report Logs

The only usage data available to report configurations is data that has been rotated. To ensure the most recent data is available for configurations, rotate the report log manually by clicking the Rotate Report Log button on the Configure and Manage Vendor Daemon screen. Depending on your license usage, you should wait a few hours or a few days after enabling report logging before rotating the report log to allow a significant amount of usage data to accumulate.

Report Log Administration

Each time a report log is rotated, an archive report log file is copied to the machine on which the FlexNet Manager for Engineering Applications server resides, and this archive report log becomes available to report configurations. A copy is also retained on the FlexNet Agent machine in the data directory for the FlexNet Agent installation.

FlexNet Manager for Engineering Applications archive logs are stored on the license server machine, with file names in the following format:

`<server_name>-<vendor_name>-yyMMddhhmmssSSS-<sequence_number>.r1`

Table 2-2 • Report Log Name Components

| Component | Description |
|--------------------------------------|--|
| <code><server_name></code> | Name of the license server host. |
| <code><vendor_name></code> | Name of the vendor daemon. |
| <code>yyMMddhhmmssSSS</code> | Timestamp (in 24-hour format) corresponding to the time at which the report log is rotated by the vendor daemon, local to the system that originally created the report log. |
| <code><sequence_number></code> | A running sequence number that the FlexNet Agent appends to the file name. This ensures uniqueness if the report log is rotated twice within the same second. |

For example, consider the archive report log with the name:

`TestLicenseServer1-avend10-131103104137-401.r1`

This file was written on November 3, 2013 at 10:41:37 by the vendor daemon avend10 on the license server TestLicenseServer1.

Rotated report logs are stored with unique file names on the machine on which FlexNet Manager for Engineering Applications resides. Each time the active report log is rotated, a new archive file is created.



Note • Report log archives accumulate to a sizable quantity over time on the license server machine. Consider deleting older archive files that are no longer needed.

Sample Report Log Data

The FlexNet Manager for Engineering Applications installation includes a sample report log file, `sample.r1`. This file contains sample usage data that you can use to generate reports. The `sample.r1` file is located in the following directory:

`<manager_admin_install_dir>/site/samples`



Task

To use the sample data in reports:

1. Click the Reports tab, then click Report Logs to display the Report Logs page.
2. In the Import Report Log field, navigate to the `sample.r1` report log.

3. Click Import.



Note • If you use the *sample.rL* data to generate a Summary Barchart report, the number of data sets may exceed the maximum number allowed in this report type.

Report Log Timestamps and Time Zones

Timestamps in a report log are determined by the clock on the license server machine. The times are converted from the time zone of the license server machine and stored in the report log in Greenwich Mean Time (GMT), also known as Universal Time (UTC). Therefore, the data in a report log is insensitive to time zones, and you can generate a report using report logs that were written on machines in different time zones without accounting for time differences. Reports show time using the time zone set on the machine on which the report is generated. FlexNet Manager for Engineering Applications reporting translates times in UTC from the report log using the offset for the time zone of the machine on which FlexNet Manager for Engineering Applications reporting is running.

For example, if you have an event in a report log that was written at 1:00 p.m. by a vendor daemon running on a license server in the US Eastern time zone, its timestamp in the report log would be written as 6:00 p.m. GMT. If an event in a report log was written at 10:00 a.m. by a vendor daemon running on a license server in the US Pacific time zone, its timestamp would also be written as 6:00 p.m. GMT. If those two report logs were used to generate a report in the US Pacific time zone, those two events would both be reported at 10:00 a.m. If the same report were generated on a computer in the US Eastern time zone, the events would both be reported at 1:00 p.m.

Occasionally, times in reports may differ from what you expect. These differences may be caused by one of the following scenarios:

- The time zone on the license server machine is set incorrectly. If the time zone is not set correctly and the system clock has been set to show local time, this introduces an error into the UTC timestamps recorded in the report log.
- The time zone on the machine where report is running is set incorrectly. Even if the clock time looks correct, the time zone may not be set correctly.

Specifying Post-Processing Options

You can configure a script that can be called when a report completes. The post-processing scripting (command-line option) support provides a way to enter a single scripting expression in the FlexNet Manager for Engineering Applications interface. This enables typical post-generation steps—such as user ID replacement and customized formatting—to be directly invoked after the report has been produced.

Report post-processing options are available in the Output Properties screen of the report configuration wizard.

The last three settings on this screen (Report Output Directory, Output Directory Options, and Post-Processing Command) are relevant for post-processing functionality. The first two settings—Report Output Directory and Output Directory Options—deal with the configuration of the external copy of the generated report, and the Post-Processing Command setting is where you specify post-processing command-line parameters.

Report Output Directory

The Report Output Directory setting specifies a directory location for the file system accessible by FlexNet Manager for Engineering Applications. This directory is called the *base directory*.

In the Report Output Directory text field, you must specify an absolute path for the directory location; and the directory that you specify must already exist. In addition, the directory needs to have write permission so the report output can be copied there.

Output Directory Options

The **Output Directory Options** setting enables you to select one of three output modes for the report output, with regard to the directory specified in the **Report Output Directory** field.

- If the Overwrite option is selected, FlexNet Manager for Engineering Applications will attempt to delete all of the files (although not any subdirectories) in the output directory before it writes to the directory.
- With the option using time-stamped subdirectories, FlexNet Manager for Engineering Applications will not delete any files. For each report run, FlexNet Manager for Engineering Applications creates a new directory with a timestamp as the directory name and copies the generated report to the new subdirectory. For example, for a report run on June 3, 2006, at 10:08:53, FlexNet Manager for Engineering Applications creates a subdirectory named “060603100853.”
- With the option using report names and time-stamped subdirectories, FlexNet Manager for Engineering Applications will not delete any files. For each report run, FlexNet Manager for Engineering Applications creates a new directory with the report name as the directory name and the timestamp as the subdirectory name, and then copies the generated report to the new subdirectory. This way, the output is sorted in the file system by the report configuration. For subsequent reports run from the same report configuration, FlexNet Manager for Engineering Applications will copy the reports to the same report name directory, but will create a new timestamp directory for each report. For example, for a report run from a report configuration named “Monthly Summary Barchart” at June 3, 2006, at 10:08:53, FlexNet Manager for Engineering Applications creates a directory named “Monthly Summary Barchart” and a subdirectory named “060603100853.”

You must specify one of the Output Directory Options in order to activate the external copy of the report.

Post-Processing Command

The external report copy is independent of the post processing, but if in reaction to the completed and copied report, you want to execute a script or some other program, you can populate the Post-Processing Command field. In this field, you can specify the command that FlexNet Manager for Engineering Applications should execute after the report has been generated. This command typically includes the path of an executable (which can be a script processor) together with some input to properly process the report.

In many cases, the post-processing script needs additional input parameters to efficiently process the report. This input can include the location of the output directory, the list of files produced for the report, the type of report produced, and so on.

FlexNet Manager for Engineering Applications provides a number of command-line parameters that you can use for post processing. These input parameters are added to the script with a dollar sign (\$) prepended to the keyword and at run time replaced by the real values. For example, the expression—

```
/usr/bin/perl /opt/scripts/postproc.pl $outputDir $outputFiles
```

—will be emitted as the command line—

```
/usr/bin/perl /opt/scripts/postproc.pl /opt/reportOut MonthlyRaw.rif
```

—if the output directory is set to “/opt/reportOut” and the report is a raw report named “MonthlyRaw.”

Command-Line Parameters

The following parameters are available for use in the **Post-Processing Command** text field. All command-line reporting must be run from the FlexNet Manager for Engineering Applications Admin installation directory.



Note • Many parameters apply only to a limited set of report types. You should use parameters that are applicable to the report type you are defining. Using parameters not applicable to the report type will result in indeterminate values at run time.

Table 2-3 • Command-Line Parameters

| Parameter Name | Description | Report Configurations | Corresponding Report Configuration Fields | Values |
|----------------|---|-----------------------|---|--|
| \$outputDir | Name of the output directory of the external copy of the generated report | All | Report Output Directory | The report output directory should be accessible from the installation of FlexNet Manager for Engineering Applications Reporting. Enter a location; test to validate that spaces in the location are correctly substituted. |
| \$outputFiles | Comma-separated list of names of output report files | All | None | Dependent on report type. Note that the <i>ReportConfigurationName.fns</i> file will always be present, containing the report generation log. |
| \$errorStatus | Indicates whether a report ran successfully | All | None | An error number of 0 indicates that the report ran successfully; all other numbers indicate an error. |
| \$errorLine | Error description (can be empty) | All | None | Contains the error description when \$errorStatus<>0; otherwise contains an empty string "". |

Table 2-3 • Command-Line Parameters

| Parameter Name | Description | Report Configurations | Corresponding Report Configuration Fields | Values |
|-----------------|--|---|--|--|
| \$reportType | The report type that was run | All | None | One of the following: <ul style="list-style-type: none"> • FLEXnet.Usage.Raw • FLEXnet.Usage.ServerCoverage • FLEXnet.Usage.Summary • FLEXnet.Usage.Efficiency • FLEXnet.Usage.SummaryBarchart • FLEXnet.Usage.HighWaterMark • FLEXnet.Usage.OverTime |
| \$user | The FlexNet user who ran the report | All | None | Example: admin |
| \$features | Comma-separated list of features included in the report | All | Features selection | Example: feature1,feature2 |
| \$servers | Comma-separated list of license servers included in the report | All | License Server Hosts on screen 1 of the report wizard | Example: host1,host2 |
| \$vendorDaemons | Comma-separated list of vendor daemons included in the report | All | Vendor Daemons | Example: avend,bvend,cvend |
| \$startDate | Start date of the report | All | Date/Time Range | Example: 8/1/06 12:00 AM |
| \$endDate | End date of the report | All | Date/Time Range | Example: 8/1/06 12:00 AM |
| \$showZero | Indicates where unused features are included | High Water Mark, Summary Barchart, Usage Over Time, Usage Summary | Usage setting with Show Zero Usage selected (Summary Options screen) | True or False; dependent on whether this option was selected in the FlexNet Manager for Engineering Applications interface |
| \$filter | Comma-separated list of filter expressions | Raw | Event Types (Filter Properties screen) | Used, Queued, Denied, and Exceptions (dependent on selections made in the FlexNet Manager for Engineering Applications interface) |

Table 2-3 • Command-Line Parameters

| Parameter Name | Description | Report Configurations | Corresponding Report Configuration Fields | Values |
|--------------------|---|---|---|--|
| \$eventFilter | Comma-separated list of event filters | All | Filters specified in the Filter Properties screen | Example: User=="jdoe",feature=="feature1" |
| \$delimiter | Field delimiter on reports | All reports except Summary Barchart and Server Coverage | Output Format (wherever the output format can be represented as RIF) | The delimiter can be a space, tab, or any other value. |
| \$showAvail | Indicates whether a group of available licenses is also on graphs (Usage Over Time) | High Water Mark, Usage Over Time | Additional Options: Show Available Options setting selected (Output Options screen) | True or False; dependent on whether this option was selected in the FlexNet Manager for Engineering Applications interface |
| \$elimFalseDenials | Denial elimination algorithm is turned on and has the specified window | Raw, Usage Summary | Denial Filtering: "Eliminate spurious denials" option (Filter Properties screen) | True or False; dependent on whether this option was selected in the FlexNet Manager for Engineering Applications interface |
| \$sortOrder | Sort order of the records on the report | All | Sort Order on Output Options screen | f or fb or empty |
| \$featureUniq | Mode of feature identification | All | Value in "Describe Features by" field (Feature Selection screen) | name, name_ver, name_asset, name_key |
| \$timeInSeconds | Times reported in UNIX seconds (for GMT values) | Raw | Output Format: "Time as Seconds" set to Yes (Output Properties screen) | True or False; dependent on whether this option was selected in the FlexNet Manager for Engineering Applications interface |

Parameters that can have an empty value (\$features, \$servers, \$vendorDaemons, \$filter, \$eventFilter) should be used only in cases where the corresponding report configuration values are not empty. If the report configuration setting does not contain a value for a specified parameter, no argument will be emitted, which can cause the argument interpreter in the post-processing script to confuse the next argument with the value of the current argument.

Example: The list of servers is specified (\$servers), but in the License Server Host selection on the Naming page (page 1) of the Report Configuration Wizard, none has been selected (indicating to run the report for all license server hosts). Then no argument will be emitted by the post-processing invocation. If another parameter is specified, the argument evaluation logic can easily mistake the value for the next argument as the list of servers.

After you specify the post-processing expression, you can complete the report configuration. When you finish the report configuration, FlexNet Manager for Engineering Applications validates the external output directory and the post-processing expression.

After saving the report configuration, you can run the report. FlexNet Manager for Engineering Applications will generate the report, copy the report to the external directory, and then invoke the post-processing script.

Testing Post-Processing Scripts

It is recommended that you use a test script first to see the command-line arguments that are being passed in and to compare them to the expected values. The following code sample shows a simple perl script which will emit the argument values into the report log file, which can be retrieved from the Report Inbox.

```
print "Post-processor command line arguments:\n";
foreach $arg (@ARGV) {
    print "\t$arg\n";
}
```

When you understand the command-line parameters, you can adjust your existing post-processing scripts or programs to interpret the report data. All post processing scripts must be accessible from the FLEXNet Manager Reporting installation.

Error Scenarios

There are two different error scenarios—report generation failure and post-processing failure.

Report Generation Failure

In the first scenario, the report generation in FlexNet Manager for Engineering Applications fails for some reason. In this case, the `$errorNum` parameter contains a nonzero value and the `$errorLine` parameter contains a description of the problem. In most cases, you can analyze the report generation log file (a file named *ReportConfigurationName.fns* located in the output directory) to troubleshoot the problem. The state of the report in the Report Inbox will show **Failed**.

Post-Processing Failure

In the second scenario, the post processing fails and returns a non-zero exit status. (This is the return value of the process). The state of the report in the Report Inbox will also show **Failed**. This will not allow access to the data FlexNet Manager for Engineering Applications successfully produced.

Report Post-Processing Examples

Example 1: Monthly Extraction of Raw Data

Use Case: FlexNet Manager for Engineering Applications should provide the Raw data format for the managed licensing system on a monthly basis. A nightly raw report is run on the first day of the next month to extract the data. The post processing includes replacing of the user IDs with full User names and department information.

Report Configuration: Define a raw report configuration with the following settings.

1. Set the Date/Time Range to “Prior Month,” scheduled at the first of each month. Because the raw report contains data for the entire system, all of the server hosts, vendor daemons and features are included.
2. The Event Types (on the Filter Properties page) can be restricted to Used, Queued and Denied, because exceptions are not relevant to the further processing.
3. On the Raw Report Output Properties screen, select the RIF format. Note that for this format, there is no parameter because you can deduce the report output format based on the file endings of output files.
4. Select “Time as Seconds” and specify the appropriate location in the Report Output Directory field. Also, select the “Overwrite the content” option (if the output after post processing does not need to be preserved) and specify the post-processing command (in this example, Perl on UNIX is used):

```
/usr/bin/perl /opt/cblecken/perl/rawProcess.pl $outputDir $outputFiles $errorStatus $errorLine
$reportType $user $startDate $endDate $eventFilter $delimiter $elimFalseDenials $sortOrder
$featureUniq $timeInSeconds
```

These are all of the parameters that apply to a Raw report. As noted previously, because none of the servers host, vendor daemons or features are selected, the associated parameters \$servers, \$vendorDaemons, and \$features are not included in the post-processing command expression. They should be included only if a non-zero selection has been made.

5. Next, the Raw Report Fields page lists the fields you can select to include in the Raw report. The list of Raw report fields is not included in the list of parameters, because the Raw RIF format will describe the number of columns and the name of the columns in the report itself and it is more consistent to take the column numbers and column names from the report.

Report Generation: Given the sample perl script from the previous code sample (which writes the parameters to stdout), generating a report from this configuration leads to the following invocation on the command line as described in the following table.

Table 2-4 • Post-Processor Command-Line Arguments

| Argument | Description |
|-------------------------------|--|
| /opt/username/extout | Output directory |
| Raw_Sample.txt,Raw_Sample.fns | Output files; the .fns file contains the debug log |
| 0 | Report generation successful |
| "" | No error message |
| FLEXnet.Usage.Raw | Indicates Raw report |
| Admin | The user running the report |
| 3/1/06 12:00 AM | Report start date |
| 3/26/06 12:00 AM | Report end date |
| used,queued,denied | The event type filters; all selected except Exceptions |

Table 2-4 • Post-Processor Command-Line Arguments

| Argument | Description |
|----------|--|
| | The delimiter; not visible because it is a space |
| true | False denials enabled |
| f | Sort by feature |
| name | Distinguish features by name |
| true | Dates are recorded in UNIX seconds |

Example 2: Month-to-Date High Water Mark

Use Case: Every day the aggregated High Water Mark report is calculated for the entire month up to today. This report should include only a subset of vendor daemons (for this example, the two vendor daemons “exam1md” and “exan1md”). This report is run every night.

Report Configuration: Define a High Water Mark report configuration with the following settings.

1. Set the Date/Time Range to “Month to Date,” scheduled to run every night. Also select the vendor daemons “exam1md” and “exan1md” from the Vendor Daemons selection box. Do not select anything for the license server hosts or the features (on the Feature Selection page).
2. On the Summary Options page, select the “Show Zero Usage” option.
3. On the Output Options page, select the RIF and the Image output format. In the Additional Options section, select the “Show Available Licenses” and “Auto-Reduce Large Datasets” options.
4. Specify the appropriate location in the Report Output Directory field. Also, select the “Overwrite the content” option (if the output after post processing does not need to be preserved) and specify the post-processing command (in this example, Perl on UNIX is used):


```
/usr/bin/perl /opt/cblecken/perl/hwmProcess.pl $outputDir $outputFiles $errorStatus $errorLine $reportType $user $vendorDaemons $startDate $endDate $showZero $showAvail $sortOrder $featureUniq
```
5. On the Report Scheduling page (available by clicking the **Schedule** button for the report from the Report Configurations page), schedule the report to run on the first of each month.

Report Generation: Given the sample perl script from the previous code sample (which writes the parameters to stdout), generating a report from this configuration leads to the following invocation on the command line as described in the following table.

Table 2-5 • Post-Processor Command-Line Arguments

| Post-Processor Command-Line Arguments | Description |
|---------------------------------------|------------------|
| /opt/username/extout | Output directory |

Table 2-5 • Post-Processor Command-Line Arguments

| Post-Processor Command-Line Arguments | Description |
|---|--|
| High_Water_Mark_Sample2.png,High_Water_Mark_Sample0.png,High_Water_Mark_Sample1.png,High_Water_Mark_Sample3.png,High_Water_Mark_Sample4.png,High_Water_Mark_Sample5.png,High_Water_Mark_Sample6.png,High_Water_Mark_Sample0.htm,High_Water_Mark_Sample1.htm,High_Water_Mark_Sample2.htm,High_Water_Mark_Sample3.htm,High_Water_Mark_Sample4.htm,High_Water_Mark_Sample5.htm,High_Water_Mark_Sample6.htm,High_Water_Mark_Sample_toc.htm,High_Water_Mark_Sample.fns | Output files; the .fns file contains the debug log |
| 0 | Report generation successful |
| "" | No error message |
| FLEXnet.Usage.HighWatermark | Indicates High Water Mark report |
| Admin | The user running the report |
| examlmd,exanlmd | Vendor daemon list |
| 2/1/96 12:00 AM | Report start date |
| 10/20/06 12:00 AM | Report end date |
| false | Do not show zero regions |
| true | Show the available licenses |
| f | Sort by feature |
| name | Distinguish features by name |

Bulk Importing Existing Usage Data

To import existing usage data, use the `bulkimport` sub-command of the `flexnet_man_dir/flexnet` utility. This command imports report logs that you may have accumulated prior to installing FlexNet Manager for Engineering Applications, and catalogs the report logs for access by this version of FlexNet Manager for Engineering Applications.

- The user of this command must be a valid FlexNet Manager for Engineering Applications user (not an LDAP user). The user must be assigned a role with the Bulk Import Report Logs permission.
- The `bulkimport` command must be run from the FlexNet Manager for Engineering Applications Admin installation directory.

Usage

```
flexnet bulkimport -username -ppassword -ffolder [-ccustomerid] [-sserver] [-r]
```

The parameters in [brackets] are optional.



Note • If the value for the `-username`, `-ppassword`, or `-ffolder` parameter contains a comma, semicolon, or ampersand (&), enclose the value in double quotes ("").

Table 2-6 • bulkimport Parameters

| Parameter | Description | Required |
|---|--|----------|
| <code>-u</code> <i>user</i> | Specifies the FlexNet Manager for Engineering Applications user to run this command. The user must be assigned a role with the Bulk Import Report Logs permission. Enclose the value in double quotes (") if it contains a comma, semicolon, or ampersand (&). | |
| <code>-p</code> <i>password</i> | Specifies the password for user. For user admin, the default password is admin. Enclose the value in double quotes (") if it contains a comma, semicolon, or ampersand (&). | |
| <code>-f</code> <i>folder</i> | A directory containing report logs, with .r1 extensions. Specify an individual report log by providing the report log file name. Enclose the value in double quotes (") if it contains a comma, semicolon, or ampersand (&); and/or if the folder path contains spaces. | |
| <code>-c</code> <i>customerid</i> | Specifies the organization ID of the organization from which the reports in the import folder, denoted with the <code>-f</code> argument, were sent. The default value is MYORG. | |
| <code>-sserver</code> | Specifies the URL for the FlexNet Manager for Engineering Applications Admin server, which must be an https connection. For example: <code>https://localhost:8445/flexnet</code> This is an optional argument. If no argument is specified, then the server is assumed to be: <code>https://127.0.0.1:<admin server port number>/flexnet</code> | |
| <code>-ryes</code> <code>-rno</code> | Specifies whether the <i>folder</i> should be processed recursively, which means that report logs contained in any subdirectories of <i>folder</i> will be imported. If any of the values <code>true</code> , <code>yes</code> , or <code>1</code> is appended to <code>-r</code> or if no argument is specified, <i>folder</i> is processed recursively. If any of the values <code>false</code> , <code>no</code> , or <code>0</code> is appended to <code>-r</code> , <i>folder</i> is not processed recursively. | |

Example

To import all of the usage data in report logs in a particular directory, type:

```
flexnet bulkimport -uadmin -padmin -f“directory with report logs”
```

Report Types

This section explains the different types of reports that FlexNet Manager for Engineering Applications Classic Reporting can generate. FlexNet Manager for Engineering Applications Classic reports belong to three report families:

- Summary reports summarize usage information about each feature in the report logs.
- Raw reports show individual usage events in textual format.
- Server reports provide information about a license server.

Usage Over Time Report

A **Usage Over Time** report produces a line graph that shows the maximum number of licenses in use over a period of time.

The granularity of the report can be adjusted using the Summary Period menu in Step 5 of the Create Report Configuration wizard. The default summary period of None corresponds to resolution of one second, which is the same resolution found in report logs. Coarser granularities of Minute, Hour, Day, Week, Semi-Month, and Month are also available.

Usage for each feature is displayed on a separate page of the report. The horizontal axis of the line graph is time-scaled to fit all the data for one feature into one window. The vertical axis is number of licenses used; alternatively, this can be configured to show the percentage of available licenses.

For each feature, two lines are plotted: one representing licenses available and one representing licenses used. For features that have low levels of activity compared with the number of licenses available, consider omitting the available license plot from the graph. To do this, clear the **Show Available Licenses** check box in Step 5 of the Create Report Configuration wizard.

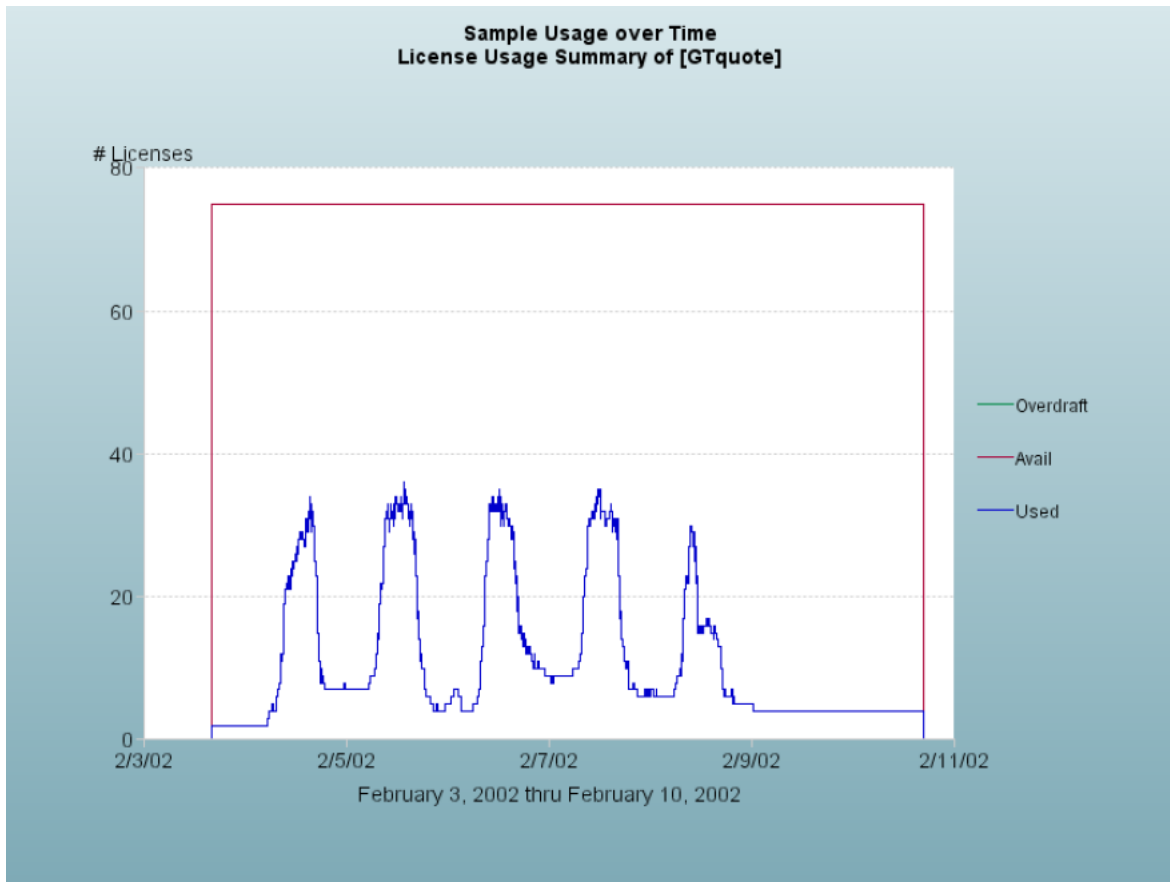


Figure 2-1: Usage Over Time report

Usage Over Time graphs are subject to automatic data reduction when data sets are particularly large. If automatic data reduction has occurred, a warning appears in the log for that report (stating that the graph contains too many data points and that the number of points is automatically reduced).

Automatic data reduction applies only to graphical reports and it reduces data density and detail. To avoid automatic data reduction, reduce the number of data points by choosing a different summary period or by narrowing the date range for the report.

High Water Mark Report

Like the Usage Over Time report, the **High Water Mark** report produces a line graph that shows the maximum number of licenses used during a selected time period. The same summary periods are available for High Water Mark graphs as for Usage Over Time graphs, but the default summary period for a High Water Mark report is `Hour`.

Each page of the report displays usage for one feature. The horizontal axis of the line graph is time, scaled to fit all the data for one feature into one window. The vertical axis is number of licenses used; alternatively, this can be configured to show the percentage of available licenses. For each feature, two lines are plotted: one, representing licenses available and the other, representing licenses used.

You can omit the plot of available licenses by clearing the Show Available Licenses option in Step 5 of the Create Report Configuration wizard.

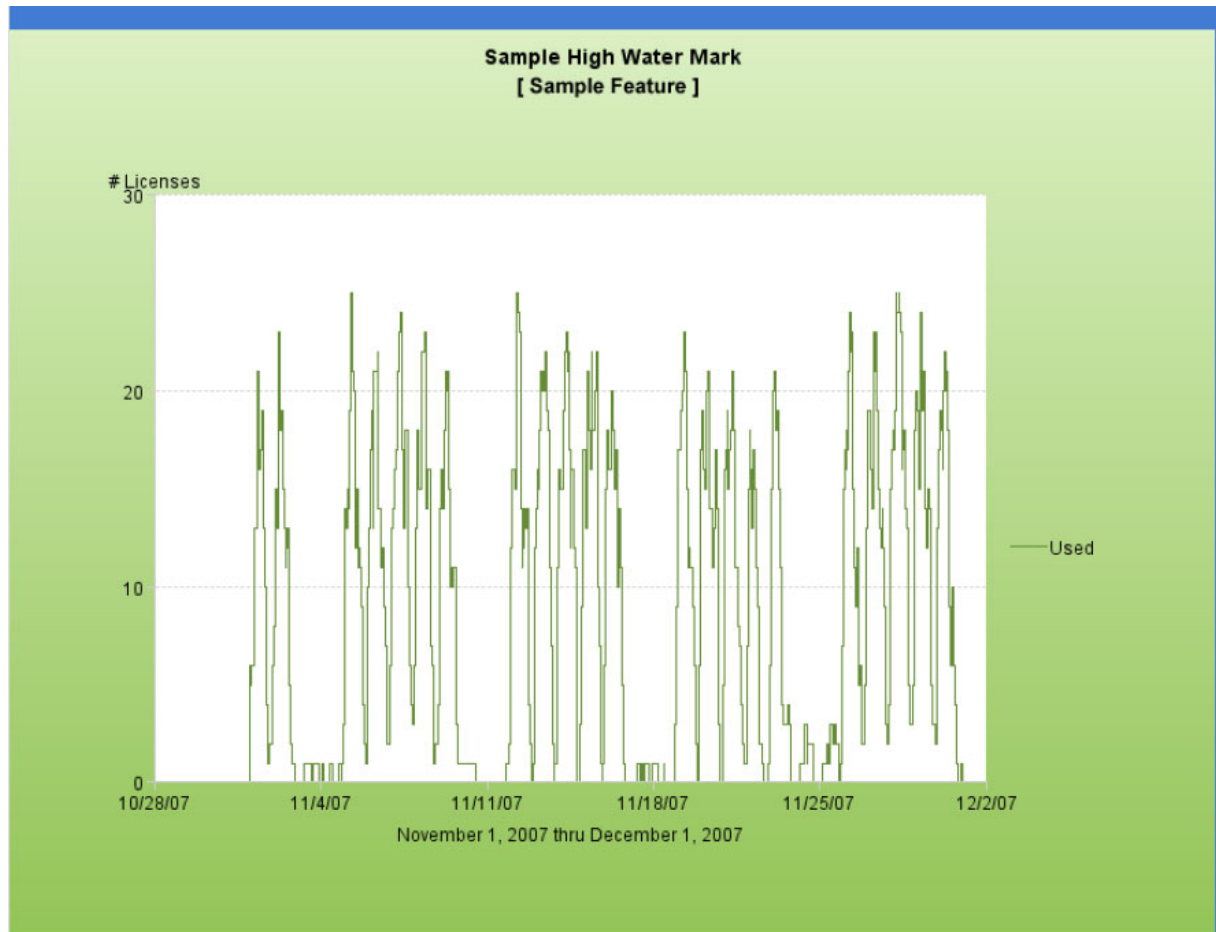


Figure 2-2: High Water Mark report

High Water Mark graphs are subject to automatic data reduction when data sets are particularly large. If automatic data reduction has occurred, a warning appears in the log for that report (stating that the graph contains too many data points and that the number of points is automatically reduced).

Automatic data reduction applies only to graphical reports and it reduces data density and detail. To avoid automatic data reduction, reduce the number of data points by choosing a different summary period or by narrowing the date range for the report.

Peak Usage and Denial Report

The **Peak Usage and Denial** report creates a bar chart report that shows the following information:

- Number of peak usage licenses within the specified summary period. You can specify a summary period with a granularity of seconds, minutes, hours, days, weeks, or months.
- Number of denials within the summary period, if there were any.
- An optional availability line, which shows the number of available licenses at a particular time.
- An optional overdraft line, which shows the number of overdraft licenses at a particular time.

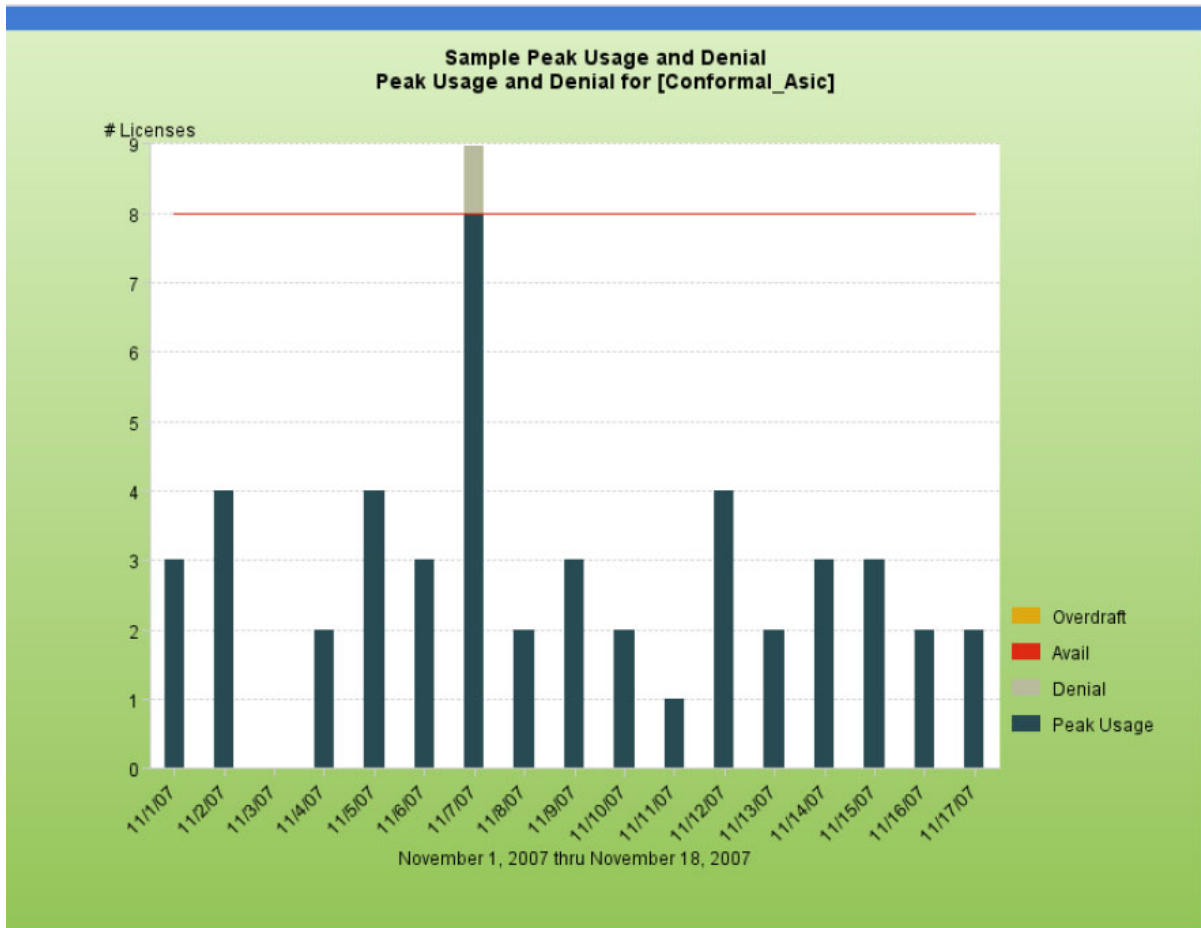


Figure 2-3: Peak Usage and Denial report



Note • There are rare cases when the vendor daemon reports unsupported events in the report log as ultimate denials instead of unsupported events. These events appear in the report log as ultimate denials, but unlike standard ultimate denial events, these events do not have any license feature associated with them. FlexNet Manager for Engineering Applications reports these events as unsupported events, rather than denials.

Summary Barchart Report

A **Summary Barchart** report provides a useful format for comparing license utilization across users, features, and so on, and is based on calculations made in a Usage Summary report. By default, a Summary Barchart shows usage for each feature that was used. License usage is displayed as bars whose heights represent one of the following metrics:

- Num Used (number of licenses checked out)
- % Used (percentage of available license hours used)
- Hours Used (number of available license hours used)
- Max Used (maximum number of licenses checked out at a time)

The bars are arranged in alphabetical order. The following figure shows an example of the percentage of available license hours used, summarized by feature. The following report compares the number of checkouts for each feature.

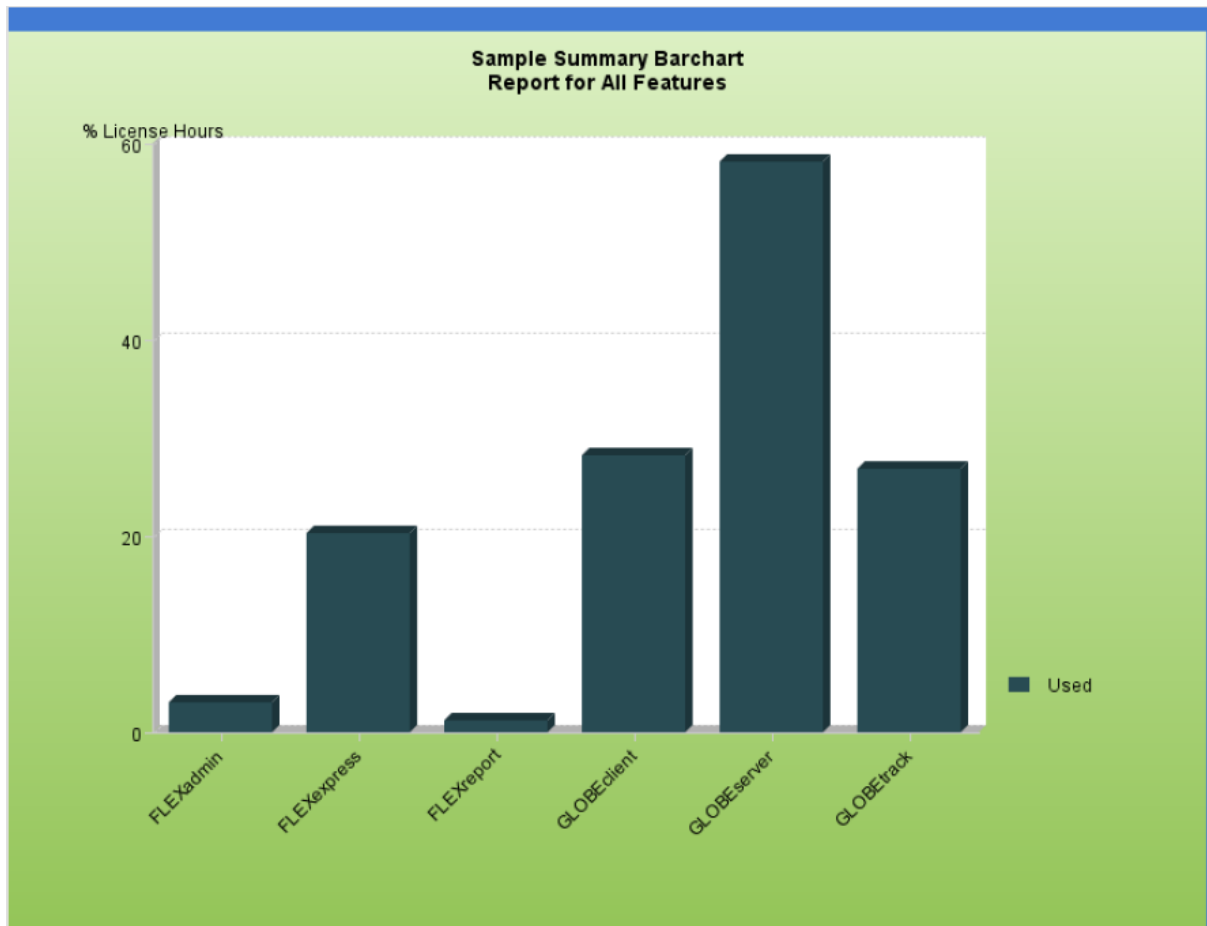


Figure 2-4: Summary Barchart report

Here is the same report of licenses used, summarized by Feature, then User. It compares the number of checkout requests satisfied for each user of a feature.

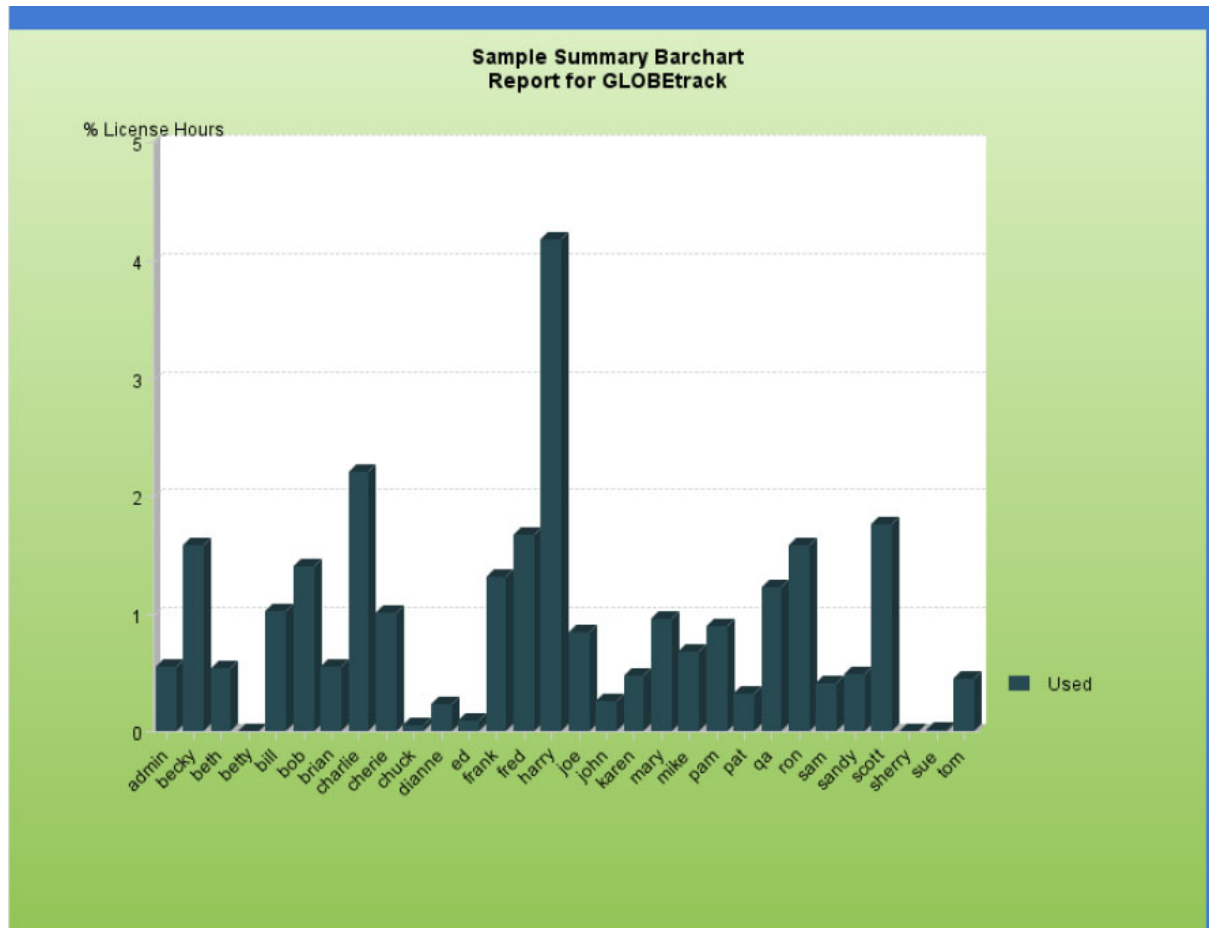


Figure 2-5: Summary Barchart report, summarized by Feature, then by User

Here is an example of the same report with license usage summarized by feature over each day in the report log. This report compares the daily usage levels for each feature over the reporting period.

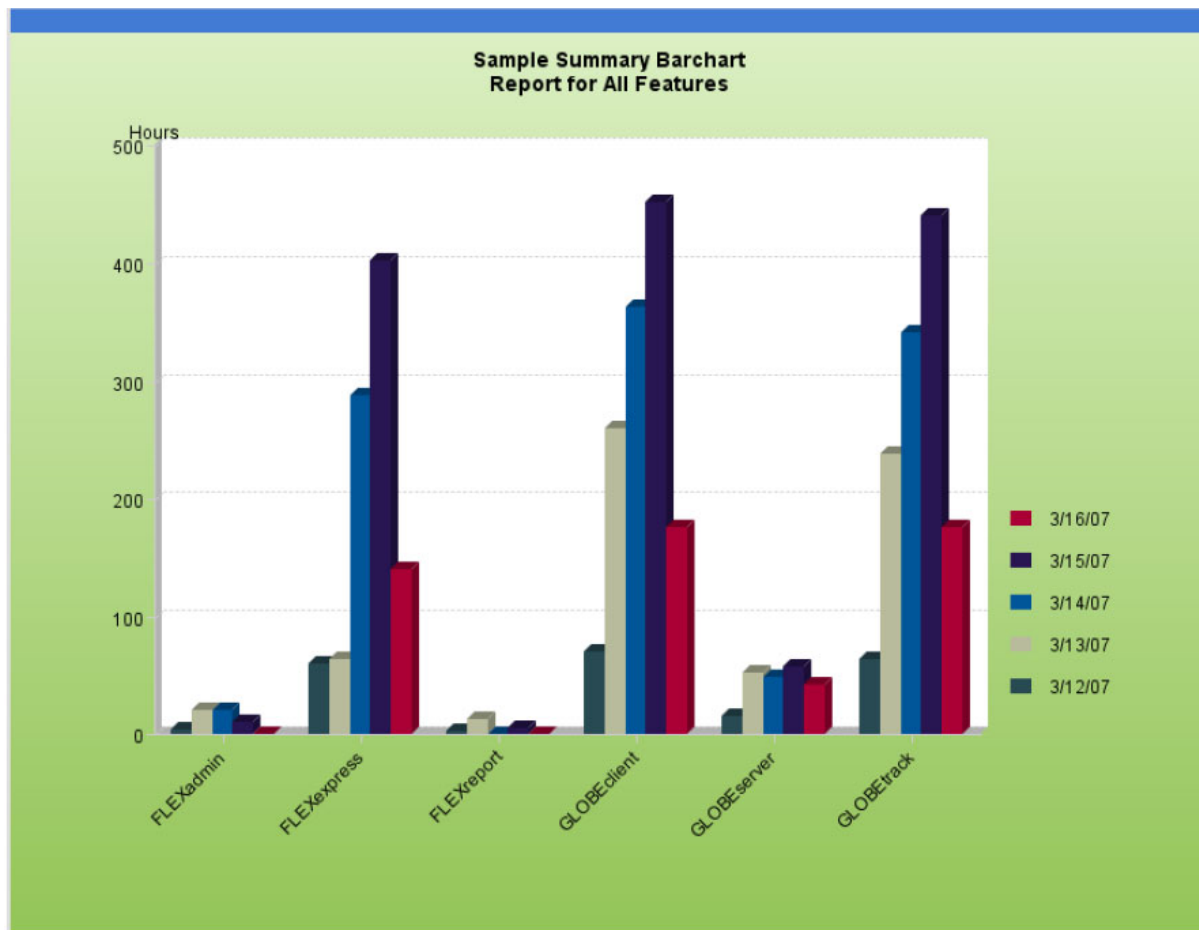


Figure 2-6: Summary Barchart report, summarized by feature over each day in the reporting period

Usage Efficiency Report

The **Usage Efficiency** report shows you how long each successive license for a particular feature was in use. This kind of analysis may help you decide how many licenses you need for that feature.

The report legend shows the maximum number of licenses checked out at one time, compared to the total number of licenses available. The bottom bar in the graph shows how long the first license for that feature was checked out, the next bar shows how long a second license was checked out, and so on, up to the maximum number of licenses checked out at one time.

The following figure shows an example of a Usage Efficiency report. It shows that only 26 out of 52 available licenses were ever in use at one time.

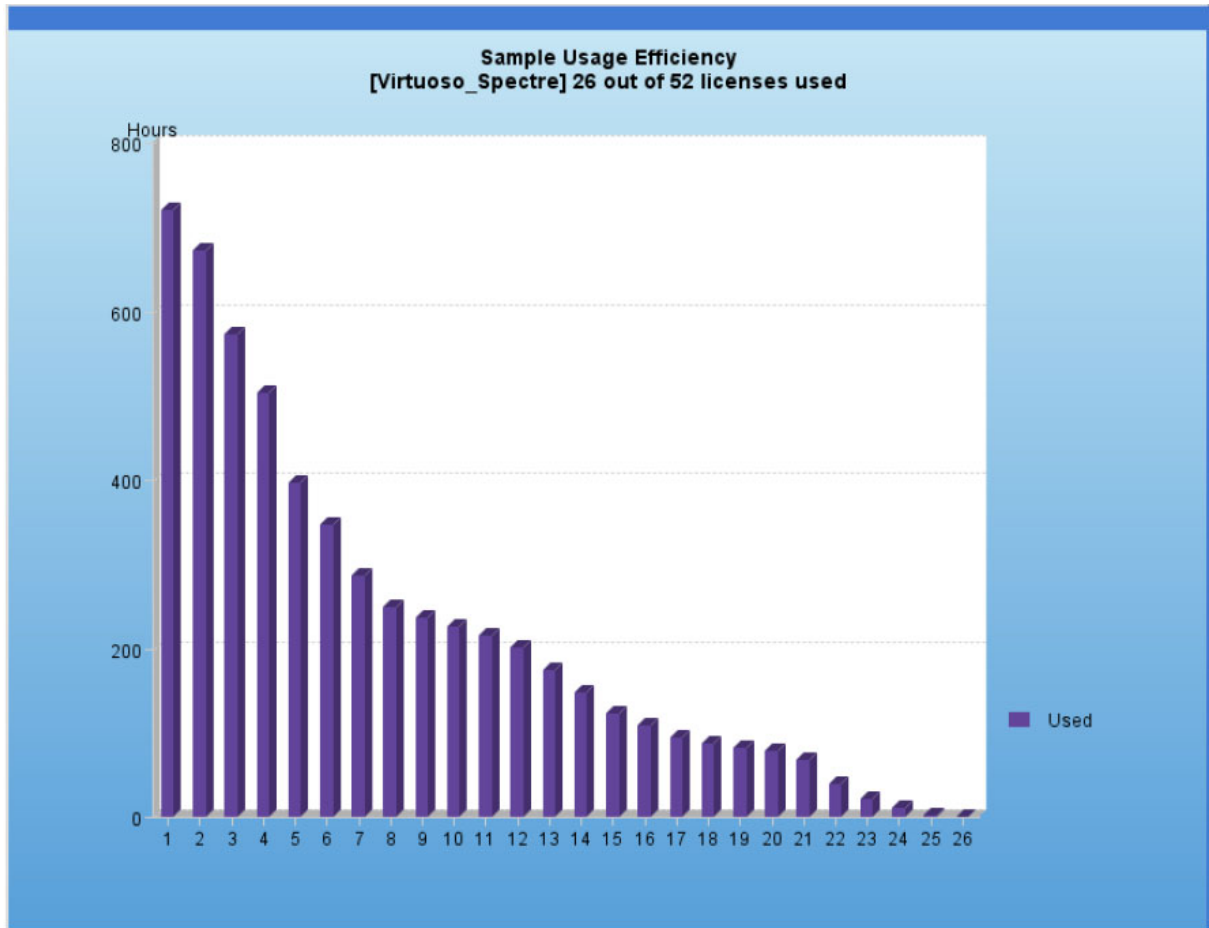


Figure 2-7: Usage Efficiency report

Several special notations may appear on this report. These notations represent unusual usage—usage of overdraft licenses and usage in excess of the license limit.

- Overdraft usage is denoted with an asterisk (*). For example, a bar marked 4* represents the amount an overdraft license is used.
- Usage in excess of the license limit is denoted with a question mark, (?). For example, a bar marked 5? represents the amount five licenses were used simultaneously if the license limit were less than five.

You can get apparent usage above your license limit in one of several ways:

- Missing or corrupted checkins in the report log
- Missing or overlapping timestamps in the report log
- Rereading an updated license file that contains new license keys—licenses checked out with the old license keys are grandfathered in, but usage of licenses with new keys starts from zero.

Usage Summary Report

A **Usage Summary** report produces textual output that summarizes the usage statistics of each feature. The output of a Usage Summary report can be saved to files in HTML, text, or RIF format.

Here is the header of a Usage Summary report in HTML format. It provides a summary of information about the report log and the parameters of the report.

| Sample Usage Summary | | Generated on Sep 29, 2009 10:02:55 PM (GMT-8.0DST60) | |
|-----------------------------|---|--|--|
| Output Format: | HTML | | |
| Start Time: | Mar 12, 2007 5:11:32 PM (GMT-8.0DST60) | Requested Start: | Feb 1, 2007 12:00:00 AM (GMT-8.0DST60) |
| End Time: | Mar 16, 2007 2:44:02 PM (GMT-8.0DST60) | Requested End: | Apr 1, 2007 1:00:00 AM (GMT-8.0DST60) |
| Vendors: | pvend10 | License Server Host(s): | CygnusX |
| Uptime: | 3 days 21 hours 32 mins 30 secs | Produced by: | FLEXnet Reporting Module v14.0.0 |
| Sort by: | feature, user | | |
| Breakdown: | user | | |
| Event Filters: | -exception-queued-used-denied | Filters: | None |
| False Denial Elimination: | 10 second threshold. Number of denials eliminated: 14 (74%) | Time Periods: | All |
| Features with activity : | FLEXadmin , FLEXexpress , FLEXreport , GLOBEclient , GLOBEserver , GLOBEtrack | | |
| Features without activity : | None | | |

Figure 2-8: Usage Summary report header

Each field in the header is explained in the following table.

Table 2-7 • Header Fields for the Usage Summary Report

| Field Name | Description |
|---------------------------------|---|
| Start Time | Date of the first usage entry in this report. This date falls within the range of Requested Start Date and Requested End Date . |
| End Time | Date of the last usage entry in this report. This date falls within the range of Requested Start Date and Requested End Date . |
| Vendors | Vendor daemons that generated the usage data for this report. |
| Uptime | The length of time during which the license server was up within the range of the Requested Start Time and Requested End Time . |
| Requested Start | The start date of the reporting period as specified in the report configuration. |
| Requested End | The end date of the reporting period as specified in the report configuration. |
| License Server Host(s) | The host names of servers on which the usage data included in the report was produced. |
| Sort By | List of categories used to sort the report; could be none. |
| Breakdown | Report breakdown categories, if any. |
| Event Filters | List of event filters for the report; could be none. |
| False Denial Elimination | Whether false denials were considered for elimination and, if so, how many were eliminated. |
| Generated on | The date on which the report was generated. |
| Produced by | The name and version information of the application that generated the report. |

Table 2-7 • Header Fields for the Usage Summary Report

| Field Name | Description |
|-------------------------------|---|
| Time Period Filters | List of time period filters for the report, if any. This field is omitted from the report if no time period filters are specified. |
| Features With Usage | List of features in the scope of this report (that is, not filtered out) that have license activity displayed in the report. |
| Features Without Usage | List of features in the scope of this report (that is, not filtered out) that do not have license activity displayed in the report. |

The following figure shows an excerpt from a Usage Summary report displayed in HTML format. It shows a summary of the usage for feature FEATURE_1, by user, and with the summary period set to Day.

| user_project user_hostname version | Time Period | Max Avail | Hrs Avail | Requests | Used | Duplicates | Queued | Denied | Unsupp | %Denied | Max | Hours | Total Used | %Used |
|---------------------------------------|-----------------|--------------|-----------|----------|------|------------|--------|--------|--------|---------|-----|---------|---------------|-------|
| All | Jun 17, 2020 | 300 | 0.0833 | 0 | 0 | 0 | n/a | n/a | n/a | n/a | 1 | 0.0003 | 0.0003 | 0.33% |
| All | Jun 18, 2020 | 300 | 5099.6667 | 17 | 17 | 0 | n/a | n/a | n/a | n/a | 1 | 16.7461 | 16.7461 | 0.33% |
| All | Total | 300 | 5099.7500 | 17 | 17 | 0 | n/a | n/a | n/a | n/a | 1 | 16.7464 | 16.7464 | 0.3% |

Figure 2-9: Usage Summary report, showing usage for feature FEATURE_1, by day

The following figure shows an excerpt from a Usage Summary report displayed in HTML format. It shows a summary of the usage for feature f1, by user, and with a summary period of Hourly.

| user_project | Time Period | Max Avail | Hrs Avail | Requests | Used | Duplicates | Queued | Denied | Unsupp | %Denied | Max | Hours | Total Used | %Used |
|--------------|--------------------|-----------|-----------|----------|------|------------|--------|--------|--------|---------|-----|--------|------------|-------|
| All | Jun 17 20, 11:59PM | 300 | 0.0833 | 0 | 0 | 0 | n/a | n/a | n/a | n/a | 1 | 0.0003 | 0.0003 | 0.33% |
| All | Jun 18 20, 12:00AM | 300 | 299.4167 | 1 | 1 | 0 | n/a | n/a | n/a | n/a | 1 | 0.7822 | 0.7822 | 0.26% |
| All | Jun 18 20, 01:00AM | 300 | 300.0000 | 1 | 1 | 0 | n/a | n/a | n/a | n/a | 1 | 0.9697 | 0.9697 | 0.32% |
| All | Jun 18 20, 02:00AM | 300 | 300.0000 | 1 | 1 | 0 | n/a | n/a | n/a | n/a | 1 | 0.9997 | 0.9997 | 0.33% |
| All | Jun 18 20, 03:00AM | 300 | 300.0000 | 1 | 1 | 0 | n/a | n/a | n/a | n/a | 1 | 0.9997 | 0.9997 | 0.33% |
| All | Jun 18 20, 04:00AM | 300 | 300.0000 | 1 | 1 | 0 | n/a | n/a | n/a | n/a | 1 | 0.9997 | 0.9997 | 0.33% |
| All | Jun 18 20, 05:00AM | 300 | 300.0000 | 1 | 1 | 0 | n/a | n/a | n/a | n/a | 1 | 0.9997 | 0.9997 | 0.33% |
| All | Jun 18 20, 06:00AM | 300 | 300.0000 | 1 | 1 | 0 | n/a | n/a | n/a | n/a | 1 | 0.9997 | 0.9997 | 0.33% |
| All | Jun 18 20, 07:00AM | 300 | 300.0000 | 1 | 1 | 0 | n/a | n/a | n/a | n/a | 1 | 0.9997 | 0.9997 | 0.33% |
| All | Jun 18 20, 08:00AM | 300 | 300.0000 | 1 | 1 | 0 | n/a | n/a | n/a | n/a | 1 | 0.9997 | 0.9997 | 0.33% |
| All | Jun 18 20, 09:00AM | 300 | 300.0000 | 1 | 1 | 0 | n/a | n/a | n/a | n/a | 1 | 0.9992 | 0.9992 | 0.33% |

Figure 2-10: Usage Summary report, showing usage for feature f1, with a summary period of Hourly

Table 2-8 describes the columns listed in a Usage Summary report.

Table 2-8 • Column headings for the Usage Summary report

| Column Heading | Description |
|------------------|--|
| Max Avail | Maximum license count available for the specified time period. When the availability for a feature is unlimited, the Max Avail is displayed as Uncounted. |
| Hrs Avail | Available license count multiplied by the time available. When the availability for a feature is unlimited, the Hrs Avail is displayed as n/a. |
| Requests | Total number of license requests: Used + Denied + Unsupp (unsupported) |

Table 2-8 • Column headings for the Usage Summary report

| Column Heading | Description |
|-------------------|--|
| Used | Number of license requests fulfilled. |
| Duplicates | Number of licenses requests where duplicate grouping was used to prevent the need to use a new license |
| Queued | Number of license requests queued. |
| Denied | Number of license requests denied. If the client application was built using FlexNet Licensing 9.0 or later, only true (ultimate) denial events by Vendor Daemon will be in the report. All that are not ultimate denial events are ignored. |
| Unsupp | Number of license requests not fulfilled because the feature is unsupported. These are also referred to as exception events. |
| %Denied | Percent of license requests that were denials. For more information, see “License Denials.” |
| Max | Maximum number of licenses used simultaneously, including overdraft licenses. |
| Hours | Number of available license hours used. |
| Total Used | Total number of license hours used. Unlike Hours , this includes license hours where the usage was reduced due to duplicate-grouping, and as such may be greater than Hrs Avail . |
| %Used | Percent of available license hours used. The formula is: $\% \text{ Used} = \text{Used hours} / \text{Available hours} * 100$ |



Note • For more information on how duplicate grouping can affect license usage, review “Duplicate Grouping” in the concepts guide.

Denials in the Usage Summary Report

Possible explanations for denials in reports where the maximum available number of licenses was never used include:

- **The number of available licenses changed during the report period.**

For example, you start your report log when you have 10 licenses of FeatureA, then a few months later, still writing to the same report log, you add 10 more FeatureA licenses. Your report might show the max available licenses = 20, max used licenses = 18, and some denials. Remember that the denials may have occurred when you only had 10 licenses available.

- **A vendor enabled the “use all features” switch.**

If this switch is enabled and you try to check out a license for a feature that has multiple feature lines, some of which are already checked out, denials are logged for each feature line that is already in use until the license server comes to a license that is available.

- **Licenses are reserved for particular users.**

Even though all licenses for a feature are not checked out, they are not available to all users. If someone outside the reservation group tries to check out a reserved license, that user will be denied a license. License reservations are set in a vendor daemon's options file.

- **Users are excluded from the group of users who can check out a license.**

Even though all licenses for a feature are not checked out, a particular user may be excluded from using a license and gets a denial when trying to check out a license. Exclusions are set in a vendor daemon's options file.

- **Total number of licenses for a given feature are divided among multiple license servers.**

When all licenses for a feature allocated to a particular license server are checked out, additional license requests from that server will be denied even though licenses are available for the same feature from another server.

Unique User Report

A **Unique User** report shows the number of unique users for specified vendors and/or features during a given time period. This report enables you to see which features are most and least heavily used by your organization. You can choose to display user names in the report.

You can choose an output format of either HTML (the default) or XML.

The report can be sorted by:

- **Feature only:** the report will display unique user counts by features.
- **Vendor then Feature:** the report will display unique user counts by vendors then features.
- **None:** the report will only display one total number of unique user count.

The following figure shows an example of the Unique User report sorted by vendor and then feature. "View Users" links are added for all features and all vendors. Click the link to view the page of unique user names for the feature/vendor.

| Sample Unique User | | Generated on Sep 29, 2009 10:13:40 PM (GMT-8.0DST60) | |
|---|--|--|--|
| Output Format: | HTML | | |
| Start Time: | Jan 31, 2007 8:05:09 AM (GMT-8.0DST60) | Requested Start: | Feb 1, 2007 12:00:00 AM (GMT-8.0DST60) |
| End Time: | Apr 1, 2007 8:40:12 AM (GMT-8.0DST60) | Requested End: | Apr 1, 2007 1:00:00 AM (GMT-8.0DST60) |
| Vendors: | MLM, pvend10 | License Server Host(s): | bev1ss005, CygnusX, licusa, mios34 |
| Sort by: | Vendor then Feature | Produced by: | FLEXnet Reporting Module v14.0.0 |
| Total Unique Users: | 110 | | |
| Features without activity : Neural_Network_Toolbox, TMW_Archive, Filter_Design_Toolbox, Simulink_Control_Design | | | |

| MLM (79 users in total) | | View Users | |
|-------------------------|---------|----------------------------|--|
| Feature | # Users | | |
| Communication_Blocks | 3 | View users | |
| Communication_Toolbox | 5 | View users | |
| Control_Toolbox | 4 | View users | |
| Data_Acq_Toolbox | 1 | View users | |
| Fixed_Point_Toolbox | 7 | View users | |
| Fixed-Point_Blocks | 4 | View users | |
| Instr_Control_Toolbox | 1 | View users | |

| MLM (continued) | | | |
|----------------------|---------|----------------------------|--|
| Feature | # Users | | |
| MATLAB | 78 | View users | |
| Optimization_Toolbox | 4 | View users | |
| Signal_Blocks | 8 | View users | |
| Signal_Toolbox | 24 | View users | |
| SIMULINK | 10 | View users | |
| Statistics_Toolbox | 5 | View users | |

| pvend10 (31 users in total) | | View Users | |
|-----------------------------|---------|----------------------------|--|
| Feature | # Users | | |
| FLEXadmin | 8 | View users | |
| FLEXexpress | 31 | View users | |
| FLEXreport | 4 | View users | |

| pvend10 (continued) | | | |
|---------------------|---------|----------------------------|--|
| Feature | # Users | | |
| GLOBEclient | 31 | View users | |
| GLOBEserver | 10 | View users | |
| GLOBEtrack | 30 | View users | |

Figure 2-11: Unique User Report - Sorted by Vendor then Feature

Server Coverage Report

The **Server Coverage** report documents the time during which a license server serves licenses. For each vendor daemon, the uptime as well as any significant periods of down time are shown. By default, only gaps in license service exceeding 15 minutes are shown on this report. You can see data for a different minimum gap in service by specifying a Coverage Gap Threshold value in seconds (on the Server Coverage Properties page of the Report Configuration Wizard).

The following figure shows an example of a Server Coverage report.

```
FLEXnet License Server Coverage Report      Sep 29, 2009 10:21:44 PM (GMT-8.0DST60)
Produced by FLEXnet Report Module v14.0.0

Report Log(s) Start: Monday, March 12, 2007 5:11:32 PM
Report Log(s) End  : Friday, March 16, 2007 2:44:02 PM

Report Start Date : Thursday, March 1, 2007 12:00:00 AM
Report End Date   : Sunday, April 1, 2007 1:00:00 AM

Filter Start Date: Thursday, March 1, 2007 12:00:00 AM
Filter End Date  : Sunday, April 1, 2007 1:00:00 AM

Gap Threshold: 15 mins 0 secs

Vendor: [pvend10]
License Server Host(s): [CygnusX]

-----
Vendor:pvend10                               Hostname:CygnusX
Hostid:000ea654ddb3
-----
Server Down Periods:
3/1/07 12:00 AM      3/12/07 5:11 PM      11 days 16 hours 11 mins 32 secs
3/16/07 2:44 PM     4/1/07 1:00 AM      15 days 10 hours 15 mins 58 secs

Uptime:  3 days 21 hours 32 mins 30 secs (12.57%)

Corrupted Blocks:
None.

Missing Blocks:
None.

-----
Report Log Processing Errors
-----
```

Figure 2-12: Server Coverage report example

In addition to reporting the time that the license server was up, this report shows:

- **Server Down Periods:** Periods of time that the license server was down
- **Corrupted Blocks:** Corrupted blocks of data in the report log
- **Missing Blocks:** Missing blocks of data in the report log
- **Report Log Processing Errors:** Errors discovered with the report log during report generation.



Note • If a start and end date filter is specified that covers a wider period of time than is represented by the report log data, then the extra width is treated as a gap and is noted in the Server Down Periods section of the report.

Raw Report

A **Raw** report shows individual usage events, not a summary of license usage. In general, a Raw report gives you one line of data for every license checkout, listed in chronological order of checkin. Raw reports can help you troubleshoot license server activity. For example, you may want to analyze the nature of certain denial or exception events.

The following figure shows an excerpt from a Raw report.

License Server Host(s): CygnusX

```
Filter Start Date : Thu, Mar 1, 2007 00:00:00 PST
Filter End Date   : Sun, Apr 1, 2007 01:00:00 PDT
Event Type Filter : -exception-queued-used-denied
Event Filters     : None
False Denial Elimination: 10 second threshold
Number of denials eliminated: 14 (73%)
```

| Event | User | Host | Display | # | Time | Duration | Group |
|-----------------------|------|-------|-------------|---|-----------------------|--------------|-------|
| Events For: FLEXadmin | | | | | | | |
| Used | fred | bee | gt:0.0 | 1 | Mar 12, 2007 17:14:58 | 00d 00:21:25 | N/A |
| Used | mary | ibm | ss4:0.0 | 1 | Mar 12, 2007 17:14:40 | 00d 00:46:56 | N/A |
| Used | fred | bee | /dev/pts/0 | 1 | Mar 12, 2007 20:22:35 | 00d 00:01:33 | N/A |
| Used | fred | bee | /dev/pts/0 | 1 | Mar 12, 2007 20:37:29 | 00d 00:00:43 | N/A |
| Used | joe | peppy | /dev/ttypl | 1 | Mar 12, 2007 20:16:30 | 00d 16:31:11 | N/A |
| Used | joe | peppy | /dev/tty7 | 1 | Mar 13, 2007 13:13:03 | 00d 01:10:52 | N/A |
| Used | fred | bee | /dev/pts/0 | 1 | Mar 13, 2007 15:09:18 | 00d 00:00:17 | N/A |
| Used | joe | peppy | /dev/tty7 | 1 | Mar 13, 2007 16:03:37 | 00d 00:00:05 | N/A |
| Used | fred | wit | /dev/pts/10 | 1 | Mar 13, 2007 16:05:10 | 00d 00:02:14 | N/A |

Figure 2-13: Raw report

The columns in the previous report are, from left to right: event type (Used, Denied, Queued, Exception), User, Host, Display, # (count), Time (checkout time), Duration (duration of checkout), and Group. You can specify the fields you want to appear in the report during Step 4 of the Create Report Configuration wizard.

For descriptions of the available fields, see the *FlexNet Manager for Engineering Applications Help Library*.

Flexera Analytics Reporting

Flexera Analytics is built on the Cognos Analytics reporting engine, and enables you to gather, collate, and report on all of your FlexNet licensing activity. Flexera Analytics includes complete tools for managing, creating, and editing both simple and complex reports.



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

FlexNet Manager for Engineering Applications provides multiple levels of usage data summarization with Hourly, Daily, and Monthly aggregations (roll-ups). These aggregations are used in the Flexera Analytics (Cognos) package 2010 Build 1 and enable you to author high-performance business reports in the Flexera Analytics while using very little extra storage space. (See [Usage Data Maintenance](#) for information about how the data cleanup service is used in conjunction with the different levels of aggregation.)

Flexera Analytics includes a detailed help facility that explains its features and functionality. In addition, you may need to refer to the Cognos documentation for both Query Studio and Report Studio, available from the Cognos web site at www.cognos.com.

For information about Flexera Analytics predefined reports and the data model, see the *FlexNet Manager for Engineering Applications Guide* for the appropriate Cognos package.

Components

Flexera Analytics has the following components:

Databases and Servers

- **Flexera Analytics:** Flexera Analytics is built on the Cognos BI (Business Intelligence) reporting engine. Flexera Analytics includes its own web server.
- **Content Store:** The *content store* is a relational database used by Flexera Analytics to store information about reporting models, folders, reports, and saved results.

Reporting Tools

- **Cognos Connection:** Cognos Connection is the Flexera Analytics portal, launched from the Reports tab, and provides access to viewing, creating and editing reports.
- **Cognos Configuration Tool:** This utility is used to set up and configure the Content Store database, Flexera Analytics components, and to start and stop Flexera Analytics services.
- **Query Studio:** Query Studio is a web-based reporting tool, used to build simple reports quickly and easily from the FlexNet Manager for Engineering Applications Reports tab.
- **Report Studio:** Report Studio is a web-based tool used by professional report authors, used to build detailed and sophisticated multiple query reports. Report Studio is launched from Cognos Connection.

Flexera Analytics Reports

You access Flexera Analytics reports using the Flexera Analytics tab beneath the Reports tab in the FlexNet Manager for Engineering Applications interface. Click the Cognos Reports link on the left side of the screen. Predefined reports are in the FlexNet Manager Suite folder.

For a complete list of the available predefined reports, along with report descriptions, see the *Flexera Analytics 2020 R1 Guide*.

Data Warehouse and Aggregation

For increased reporting efficiency, Flexera Analytics (Cognos) reports are built from data contained in a set of database tables called the *data warehouse*, which stores usage data. Aggregation is the process that parses the report log data that has been imported to FlexNet Manager for Engineering Applications and populates the data warehouse with usage data. It is sometimes referred to as a “roll-up”.

You can run aggregation on your enterprise’s report log data in two ways:

- **Incremental Aggregation**—Run aggregation for specified vendor daemons and/or for a specific time period.
- **Periodic Aggregation**—Run aggregation on a regularly scheduled, periodic basis.

For a detailed description of the FlexNet Reporting data model, see the *FlexNet Manager for Engineering Applications 2020 R1 Flexera Analytics Guide*.

For information about aggregating usage data for use in Flexera Analytics reports, see the online *FlexNet Manager for Engineering Applications 2020 R1 Help Library*.

Denial Folding

Older vendor daemons would sometimes create many denial events in response to a single request, meaning that the report log included far more denial events than users actually experienced. If FlexNet Manager for Engineering Applications were to process all of these denial events, reports that included a count of denials would show a far higher number of denials than had occurred. To avoid this higher-than-actual denial reporting, data aggregation (as well as Classic FlexNet Manager for Engineering Applications reporting) will fold these denial events.

In denial folding, all denial events for the same user for the same feature that occur within a window of time (10 seconds by default) are replaced with a single ultimate denial event, or are “thrown away” if the license is eventually granted in a usage event prior to the end of the time window. You can configure the duration of the time window using the “Aggregation denial folding interval” setting, described in the next section.

Aggregation Configuration Settings

The System Configuration | Reporting screen in the FlexNet Manager for Engineering Applications interface includes the following settings related to aggregation:

- **Aggregation denial folding interval**—Enables you to specify the window (in seconds) during which identical denials should be considered part of the same denial event.
- **Aggregation segment size**—Enables you to specify the number of days of usage data that the aggregation process will consider at one time.
- **Earliest date in data warehouse**—To reduce the size of the data warehouse, and the time the aggregation process takes, the aggregation will be limited to only consider usage data that occurs after this date. This could be useful if you have imported a large amount of data from report logs, but are interested only in reporting on the most recent data.
- **Project Parsing Regular Expression**—If your Project Name field contains data in addition to the project name, this field enables you to specify which part of the Project Name string you want FlexNet Manager for Engineering Applications to consider when usage data is aggregated. (This field is for use primarily by Platform Computing customers.)

Organizational Structures for Reporting

FlexNet Manager for Engineering Applications supports the creation of organizational structures. Organizational structures enable grouping and sub-grouping of users, projects, and license server hosts in hierarchical order. You can use organizational structures in reporting and for product license distribution.

For example, you can create organizational structures for all of the users in your organization based on the departments to which they belong. You can create another organizational structure based on the geographical location to which users belong. Both organizational structures can exist in the FlexNet Manager for Engineering Applications system at one time and you can generate reports for either group or for subgroups of these two organizational structures.

Organization structures are made up of nodes that are arranged in a tree structure. There is a single root node at the top of each organizational structure. Leaf nodes (nodes that have zero child nodes) for each organizational structure are defined by the type of the structure—for example, each leaf node of a user-based organizational structure corresponds to a user in your enterprise. A leaf node cannot have another leaf-level node as a parent. A leaf-level node cannot have more than one parent node on the same date.

The nodes between the root node and the leaf nodes are known as intermediate nodes. There can be an arbitrary number of intermediate levels in each organizational structure. A leaf-level node could have an intermediate parent node of any level.



Note • See the online *FlexNet Manager for Engineering Applications 2020 R1 Help Library* for information about defining organizational structures using the *FlexNet Manager for Engineering Applications* interface, including information about working with LDAP.

Defining Organizational Structures

You can define user-, project- or license server host-based organizational structures using the FlexNet Manager for Engineering Applications interface or you can define the organizational structures in an XML file and import the file. In an organizational structure, the leaf nodes (users, projects, or license server hosts) can be grouped into organizational unit nodes (OrgNodes) which in turn can be grouped under other OrgNodes or directly under the root organizational structure

(OrgStructure). Each organizational structure has a unique name—for example, All-Employees or all-hosts-by-geographical-location. The organizational structure is associated with a start date. You can run reports for leaf nodes, organizational unit nodes, or entire organizational structures.

All the users, projects, and license server hosts that do not belong to any organizational structure (for example, new users, projects, or license server hosts) are of UNDEFINED type. FlexNet Manager for Engineering Applications provides a default organizational structure called All Users. When you run data aggregation, all of the users that have usage will be grouped under UNDEFINED node which is a child of the All Users organizational structure. You can use this organizational structure to organize the users into the appropriate hierarchy.



Note • Aggregation performance is highly dependent on the number of organizational structures defined in the FlexNet Manager for Engineering Applications system.

Defining Organizational Structures Using an XML File

FlexNet Manager for Engineering Applications provides the XML schema for the organizational structure `orgStructure.xsd` and sample XML files in the FlexNet Reporting installation directory at `\site\samples`. The schema ensures that an organizational structure contains only one type of entities (leaf-nodes)—that is, a host organizational structure can never contain users or projects.

The organizational structure schema enables you to contain one or more structures in a single XML file at the time of import. The organizational structures can be one of three types—`userOrgStructure`, `hostOrgStructure`, or `projectOrgStructure`. The `userOrgStructure` can contain at least one or more `userOrgNodes` or one or more users. The `userOrgNode` should contain at least one more user or one or more `userOrgNode` as its child. The `hostOrgStructure` and `projectOrgStructure` have a similar hierarchy. See the section, “[Sample XML Organizational Structures](#).”

In an organizational structure, the leaf nodes (users, hosts, or projects) can be grouped into organization nodes (for example, `userOrgNode`), which could be grouped under other `OrgNodes` or the `OrgStructure`. The organizational structure is associated with a start date (`startDate`). Reports could be run for leaf nodes, organization unit nodes, or structures. The organizational structure is validated to ensure that a child node would never have more than one parent at a time.

See the following section, “[Special Characters](#),” for information about how to express some characters using XML.

Adding New Organizational Structures Using XML

You can add a new organizational structure using XML.

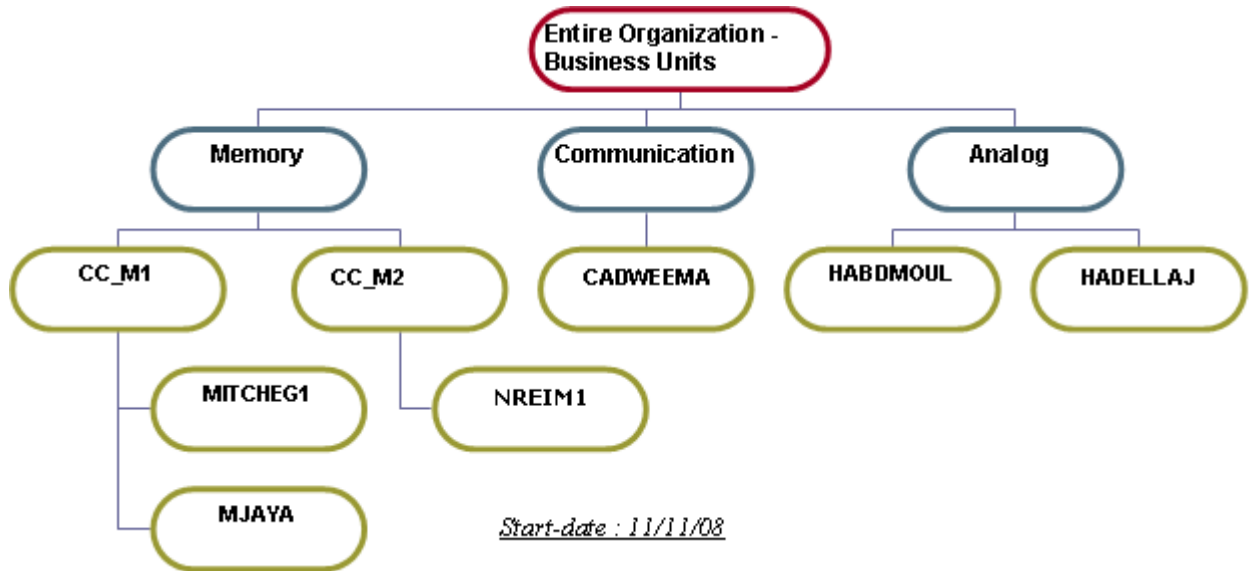


Figure 4-1: Sample organizational structure.

In [Figure 4-1](#), 'Entire Organization - Business Units' is an organizational structure with the type 'All-users' which includes only users. 'Memory', 'CC_M1', 'CC_M2', 'Communication' and 'Analog' are the org-unit-nodes. The rest of the nodes all are of the users.

The organizational structure is validated to ensure that a child org-node would never have more than one parent at a time.

When naming leaf nodes and intermediate nodes, keep in mind the following restrictions:

- **Uniqueness restriction**—A leaf node's name must be unique among other leaf nodes. Likewise, an intermediate node's name must be unique among other intermediate nodes. However, a leaf node can have the same name as an intermediate node.
- **Case restriction**—User leaf names and project leaf names must be in uppercase. Only the server leaf names are in lower case.

```

<?xml version="1.0" encoding="utf-8" ?>
<orgStructures startDate="2007-09-24" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="<FNMEA_Reporting_InstallDirectory>\site\webapps\flexnet\schema\remoter
eporting\orgStructure.xsd">
<userOrgStructure name="ENTIRE ORGANIZATION - BUSINESS UNITS" isNew="true">
  <userOrgNode name="MEMORY">
    <userOrgNode name="CC_M1">
      <user name="MITCHEG1" />
      <user name="MJAYA" />
    </userOrgNode>
    <userOrgNode name="CC_M2">
      <user name="NREIM1" />
    </userOrgNode>
  </userOrgNode>
  <userOrgNode name="COMMUNICATION">
    <user name="CADWEEMA" />
  </userOrgNode>
  <userOrgNode name="ANALOG">
    <user name="HABDMOUL" />
    <user name="HADELLAJ" />
  </userOrgNode>
</userOrgStructure>
</orgStructures>

```

```
</userOrgNode>
</userOrgStructure>
</orgStructures>
```



Note • The organizational structure name cannot be “User”, “Project”, “LicenseServer” or “UNDEFINED”. The intermediate node (OrgNode) or leaf node name (user, project, or host) cannot be “UNDEFINED”. These are reserved words.

Updating Organizational Structures Using XML

Using XML, you can update an organizational structure to add nodes and move child nodes from one parent node to another by modifying and re-importing the organizational structure.

When updating an existing organizational structure, the names of the existing nodes must remain the same (have the same names as in previous XML definitions). If a node is renamed, FlexNet Manager for Engineering Applications treats the two nodes as two different nodes belonging to different time periods.

Organizational structures can be updated to add a node that was previously deleted, as long as the start date for the node being re-added is not prior to the end-date for the same node. While updating the organizational structure, it is validated to ensure that no child node exists prior to its parent node. The best way to ensure the changes made to the structure are valid is to provide an appropriate start-date for a structure. The start-date for a structure can never be in the past.

In the following example, the “Entire Organization - Business Units” organizational structure is updated to:

- Add a new user CADWTHOM.
- Delete node HADELLAJ.
- Move node MJAYA from one parent node to another.

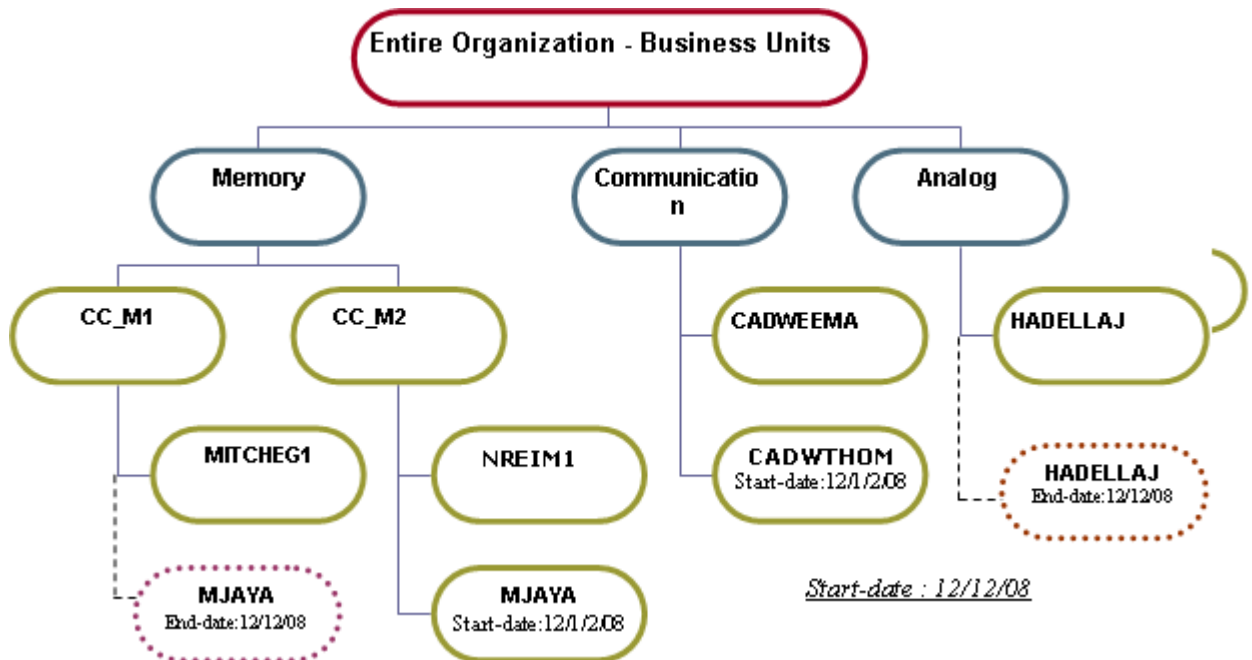


Figure 4-2: Updating the sample organizational structure.

To do this, you can re-use the XML that you imported previously. You must change the value of `isNew="true"` to `isNew="false"` and the `startDate` value may change.

```
<?xml version="1.0"?>
<orgStructures startDate="2008-12-12" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="<FNMEA_Reporting_InstallDirectory>\site\webapps\flexnet\schema\remote
reporting\orgStructure.xsd">
  <userOrgStructure name="Entire Organization - Business Units" isNew="false">
    <userOrgNode name="Memory">
      <userOrgNode name="CC_M1">
        <user name="MITCHEG1" />
      </userOrgNode>
      <userOrgNode name="CC_M1">
        <user name="NREIM1" />
        <user name="MJAYA" /> <!-- Added -->
      </userOrgNode>
    </userOrgNode>
    <userOrgNode name="Communication">
      <user name="CADWEEMA" />
      <user name="CADWTHOM" /> <!-- Added -->
    </userOrgNode>
    <userOrgNode name=" Analog ">
      <user name=" HABDMOUL" />
      <!-- deleted HADELLAJ -->
    </userOrgNode>
  </userOrgStructure>
</orgStructures>
```

After the update, the mapping between the nodes CC_M1 and MJAYA, Analog and HADELLAJ are updated to have an end date. New mappings between Communication and CADWTHOM and CC_M2 and MJAYA are added with the later start date. The rest of the mappings in the organizational structure remain unchanged.

The update process no longer requires a whole snapshot of the organizational structure. We now will update only what is defined in the XML file. The rest of the predefined relationships in the database are left untouched.

Special Characters

In the XML file, some special characters must be written in a particular way to be processed correctly.

Table 4-1 • XML Special Characters

| Character | Syntax |
|----------------------------|--------|
| & (ampersand) | & |
| > (greater than) | > |
| “ (double quote) | " |

Sample XML Organizational Structures

The organizational structure schema enables you to contain one or more structures in a single XML file at the time of import. The organizational structures can be one of three types—[userOrgStructure](#), [hostOrgStructure](#), or [projectOrgStructure](#).

Attributes of Root Node

Table 4-2 • Root Node Attributes

| | | |
|---------------------|-------|----------------|
| hostOrgStructure | name | 255 characters |
| userOrgStructure | | |
| projectOrgStructure | isNew | true/false |

Attributes of Intermediate Node

Table 4-3 • Intermediate Node Attributes

| | | |
|----------------|-------|------------------|
| hostOrgNode | name | 255 characters |
| userOrgNode | | |
| projectOrgNode | notes | 1,000 characters |

Attributes of User Leaf Node

Table 4-4 • User Leaf Node Attributes

| | | |
|------|-------------|----------------|
| user | name | 255 characters |
| | firstName | 255 characters |
| | lastName | 255 characters |
| | email | 255 characters |
| | phoneNumber | 255 characters |
| | cellNumber | 255 characters |
| | status | 255 characters |
| | address1 | 255 characters |
| | address2 | 255 characters |
| | city | 255 characters |
| | postalCode | 255 characters |
| | state | 255 characters |
| | country | 255 characters |
| | department | 255 characters |

Attributes of Project Leaf Node

Table 4-5 • Project Leaf Node Attributes

| | | |
|---------|-------------|----------------|
| project | name | 255 characters |
| | displayName | 255 characters |

Attributes of Host Leaf Node

Table 4-6 • Host Leaf Node Attributes

| | | |
|------|-------------|----------------|
| host | name | 255 characters |
| | displayName | 255 characters |

Sample User Organizational Structure

```
<?xml version="1.0" encoding="UTF-8"?>
<orgStructures startDate="2005-01-01" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="<FNMEA_Reporting_InstallDirectory>\site\webapps\flexnet\schema\remoter
eporting\orgStructure.xsd">
  <userOrgStructure name="Sample Users Org Structure" isNew="true">
    <userOrgNode name="FNM Users" notes="All users using FNM product">
      <userOrgNode name="Engg">
        <user name="jon" firstName="jon" lastName="k" address1="A Street" address2="Suite 402"
city="San Francisco" state="CA" postalCode="00000" country="USA" phoneNumber="000-0000" cellNumber="000-
000-0000" email="jon@companyname.com" department="Engg" status="active"/>
        <user name="mayu" firstName="mayu" lastName="aegie" address1="A Street" address2="Suite
402" city="San Francisco" state="CA" postalCode="00000" country="USA" phoneNumber="000-0000"
cellNumber="000-000-0000" email="maegie@companyname.com" department="QA" status="active"/>
      </userOrgNode>
      <userOrgNode name="Services">
        <user name="cathy" firstName="cathy" lastName="o" address1="A Street" address2="Suite 302"
city="San Francisco" state="CA" postalCode="00000" country="USA" phoneNumber="000-0000" cellNumber="000-
000-0000" email="cathy@companyname.com" department="Sales" status="active"/>
        <user name="rob" firstName="rob" lastName="w" address1="A Street" address2="Suite 202"
city="San Francisco" state="CA" postalCode="00000" country="USA" phoneNumber="000-0000" cellNumber="000-
000-0000" email="rob@companyname.com" department="Support" status="active"/>
      </userOrgNode>
    </userOrgNode>
  </userOrgStructure>
</orgStructures>
```

The notes attribute can contain up to 1,000 characters.

Sample License Server Host Organizational Structure

```
<?xml version="1.0" encoding="UTF-8"?>
<orgStructures startDate="2005-01-01" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="<FNMEA_Reporting_InstallDirectory>\site\webapps\flexnet\schema\remoter
eporting\orgStructure.xsd">
  <hostOrgStructure name="Sample Host Org Structure" isNew="true">
    <hostOrgNode name="North America" notes="Hosts in US and canada">
      <host name="eunomia" displayName="Eunomia"/>
      <host name="irene" displayName="Irene"/>
    </hostOrgNode>
    <hostOrgNode name="Europe">
      <host name="astrea" displayName="Astrea"/>
      <host name="hebe" displayName="Hebe"/>
    </hostOrgNode>
  </hostOrgStructure>
</orgStructures>
```

The notes attribute can contain up to 1,000 characters.

Sample Project Organizational Structure

```
<?xml version="1.0" encoding="UTF-8"?>
<orgStructures startDate="2005-01-01" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="<FNMEA_Reporting_InstallDirectory>\site\webapps\flexnet\schema\remoter
eporting\orgStructure.xsd">
  <projectOrgStructure name="Sample Projects Org Structure" isNew="true">
    <projectOrgNode name="FlexNet" notes="FlexNet projects">
      <project name="FlexNet Manager" displayName="FlexNet Manager" />
      <project name="FlexNet Operations" displayName="FlexNet Operations" />
    </projectOrgNode>
  </projectOrgStructure>
</orgStructures>
```

The notes attribute can contain up to 1,000 characters.

Importing Organizational Structure XML Files

You can import organizational structure XML files in two ways:

- Using the Import button on the Organizational Structures screen in the FlexNet Manager for Engineering Applications interface.
- Using the command-line utility `flexnet importorgstructure`.



Note • Remote organizational-structure import from the command line is not supported.



Task

To import an organizational structure XML file at the command line:

1. Ensure that FlexNet Manager for Engineering Applications Admin and FlexNet Manager for Engineering Applications Reporting are running.
2. Open a Command prompt and change to the FlexNet Manager for Engineering Applications Admin installation directory `<fnm_admin_install_dir>`.
3. Type the following command:

```
flexnet importorgstructure -u<username> -p<password> -f<orgstructure-file-name-with-path>
```

To view help for this utility, run the command:

```
flexnet importorgstructure help
```

Reporting on Usage By Project

Using organizational structures, you can enable reporting by project. This way, you can see the usage of software products by various project teams in your enterprise.

**Task****To report on usage by project:**

1. Create an organizational structure based on project names. The leaf-level nodes of the organizational structure will be project names, rather than user names.
2. Set the `LM_PROJECT` environment variable on FlexNet Publisher client applications. See the FlexNet Publisher documentation for more information.
3. Run aggregation to aggregate the usage data by project.

Defining Organizational Structures Using the FlexNet Manager for Engineering Applications Interface

You can add, define, and delete organizational structures using the **Organizational Structures** navigation section in the FlexNet Manager for Engineering Applications interface. The **Organizational Structures** screen (available from the **Reporting** tab) displays a list of all of the organizational structures that have been created using FlexNet Manager for Engineering Applications or imported from an XML file.

The **Organizational Structures** navigation section is displayed only if FlexNet Manager for Engineering Applications Flexera Analytics (Cognos) has been installed and configured.



Note • See the online *FlexNet Manager for Engineering Applications 2020 R1 Help Library* for information about defining organizational structures using the FlexNet Manager for Engineering Applications interface, including information about working with LDAP.

Usage Data Maintenance

FlexNet Manager for Engineering Applications provides functionality that enables you to automate control of usage data retention and cleanup using the Usage Data Maintenance screen. To ensure that there is no conflict between running the data cleanup service and running aggregation, note the best practices described in this section.

Usage data maintenance is not enabled by default. To enable usage data maintenance, you select the data tier for which you want to enable the cleanup service in the Usage Data Maintenance screen:

- Report Log Usage Data
- Hourly Aggregation Usage Data
- Daily Aggregation Usage Data
- Monthly Aggregation Data

For each tier selected, you specify how long data should be retained before it is deleted.

Report Log Usage Data Maintenance

The report log usage cleanup service leverages the existing reporting database cleanup service that runs periodically to remove report log data that is stored in the reporting database.

When determining what report log data to delete, FlexNet Manager for Engineering Applications checks the end date of the associated report log. If that date is past the date specified for retaining report log data (according to the setting in the Usage Data Maintenance screen), then all the report log data is deleted. If the start date of the report log is past the specified date, but the end date is not, none of the data will be deleted.

After the report log data is deleted, the report log associated with the data is deleted from the FlexNet Manager for Engineering Applications server. The report log on the FlexNet Agent machine is not affected.

To prevent inconsistent data, the report log usage cleanup service is superseded in the following instances:

- During report log import
- While a Classic report is running
- While aggregation is running

Aggregation Usage Data Maintenance

If usage data maintenance is enabled, the data cleanup service—which deletes the usage data—will be invoked in the following instances:

- At reporting server startup.
- When aggregation is finished.
- When a change is made to a Usage Data Maintenance setting and saved.

After the data cleanup service is invoked, it checks whether usage data exists for deletion and then sequentially deletes:

1. Hourly usage data
2. Daily usage data
3. Monthly usage data

For every type of usage data cleanup, one week's worth of data is deleted as one "chunk" and the data cleanup service continues deleting data until it deletes the data prior to the setting configured.

Best Practices for Usage Data Maintenance

Recall that, by default, usage data maintenance is not enabled. If your reporting system has newly imported files that were never aggregated and you are unsure how many months of data you may need for reporting, it is recommended that you leave the Usage Data Maintenance settings disabled at first.

After aggregation is finished and you have analyzed your Flexera Analytics (Cognos) reports, you can specify the appropriate hourly, daily, and monthly Usage Data Maintenance settings. These three settings can have different values based on business needs—for example, your enterprise may need to retain 12 months of monthly usage data, six months of daily usage data, and three months of hourly usage data in the system.

While enabling Usage Data Maintenance, be sure to consider aggregation scheduling. When you save the Usage Data Maintenance settings, the data cleanup service will be invoked immediately. While the usage data cleanup service is running, if a scheduled or manually triggered aggregation starts, the aggregation may try to save the usage data and the data cleanup service may try to delete the usage data for the same timeframe. This could result in undesired behavior. Before running aggregation, check the reporting `flexnet.log` to determine whether the data cleanup service is in process.

The following example demonstrates how the aggregation process works:

1. FlexNet Manager for Engineering Applications builds hourly aggregation for seven days (default configuration). The input is transactional data.
2. FlexNet Manager for Engineering Applications build daily aggregation for the same seven days. The input is hourly aggregation.
3. FlexNet Manager for Engineering Applications build monthly aggregation in a similar way, taking input from daily aggregation.

In this scenario, if you re-run aggregation for a past period of time, for example, six months from now, and you have a three-month retention policy for hourly aggregation, it may result in creating hourly aggregation for the time beyond the retention date just to support the re-aggregation for daily and/or monthly aggregation. Normally the "extra" data will be deleted later, but the best practice is to disable the usage data maintenance settings prior to aggregation that may overlap in time with the retention intervals.

It is recommended you define your data-retention time intervals such a way that there is a “buffer time interval” that ensures that the usage data for cleanup would not overlap with any possible usage data for reports or aggregations.

Usage Data Cleanup Service Log Messages

When running each type of data cleanup service (hourly, daily, or monthly), the following line appears in the log to indicate that the data cleanup service is running:

```
Starting Hourly Cleaning activity for <from-date> to <to-date>
```

When the data cleanup service is finished, it will log the following:

```
Stopping Usage Data Cleanup Service.
```

If the data cleanup service found no usage data, it logs the following:

```
Hourly cleanup activity found no data to clean up.
```


Reporting on non-FlexNet Application Usage

FlexNet Manager for Engineering Applications enables enterprises to use FlexNet Manager for Engineering Applications's reporting functionality to evaluate license usage for non-FlexNet applications. FlexNet Manager for Engineering Applications supports [reporting on IBM's LUM-licensed applications](#) and on other licensed applications, using FlexNet Manager for Engineering Applications's [common report log formatting functionality](#).

Reporting for IBM LUM

If you have purchased the separately licensed IBM LUM reporting functionality, you can use FlexNet Manager for Engineering Applications to report on LUM usage data in your enterprise. In order for FlexNet Manager for Engineering Applications to process LUM report logs, you must configure the LUM license servers to log all license events and you must issue an `i4blt` command to generate LUM report logs that are compatible with FlexNet Manager for Engineering Applications.

See the "Using License Use Management" manual from IBM, for information about generating reports with `i4blt` and configuring the LUM license server to log events.

Process Overview

The general process for preparing LUM usage data for FlexNet Manager for Engineering Applications reporting is:

1. [Configure LUM license servers](#) to log specific events.
2. [Use the i4blt command](#) to generate Standard Event Reports.



Note • Each Standard Event Report must be filtered to include data from only a single license server or vendor.

3. For legacy or other event logs containing data from multiple license servers or vendors, [use the processLUMFile utility](#) to split the logs into multiple files. This enables FlexNet Manager for Engineering Applications to process the usage data for reporting.

4. Import the report log files into FlexNet Manager for Engineering Applications using the Import functionality on the Report Logs screen or using the command `flexnet bulkimport`.

Configuring LUM License Servers to Log Events

The default configuration for IBM LUM license servers does not gather or retain enough information to be useful in reporting usage. By default, each LUM license server retains only 200 KB of log data. You must increase the log data limit to retain enough data to ensure that log data is not overwritten before the data is imported into FlexNet Manager for Engineering Applications.

By default, LUM license servers log only errors and do not log normal license usage data. In addition to logging errors, the LUM license servers need to log the following events:

- Grant events
- Release events
- Timeout events

To minimize the size of the log files (and the event reports), it is recommended that you do not log Checkin events. LUM Checkin events are not the same as FlexNet licensing check-in events. LUM Checkin events are heartbeat events where the client must “check in” or make contact with the license server periodically.

Processing LUM Report Logs Using i4blt

The LUM event logs cannot be directly imported into FlexNet Manager for Engineering Applications. The LUM `i4blt` command must be used to generate Standard Event Reports, which are then imported into FlexNet Manager for Engineering Applications. In addition, to enable FlexNet Manager for Engineering Applications to distinguish between LUM and FlexNet report log files, the LUM report log file names must end with the file extension `.lrl`.

Each Standard Event Report must be filtered to include data from a single license server and a single vendor. If you have legacy files that contain data from multiple license servers and vendors, you need to use the [processLUMFile utility](#) (described in the next section) to split the file into multiple files, each containing data from a single license server and vendor.

By default, LUM servers are configured to log only errors. Be sure that your LUM servers are configured to log all license events. See the previous section, [Configuring LUM License Servers to Log Events](#) for more information.

Syntax

The general form of the `i4blt` command to create these reports is as follows. Refer to the LUM documentation for specific information.

```
i4blt -r1 -e2,4 -n "<server_host_name>" -v "<vendor_name>" -b "<start_date>" -g "<end_date>"  
<report_file_name>.lrl
```

Splitting LUM Files Using processLUMFile Utility

You may have legacy LUM event log files that contain usage data from multiple license servers and/or vendors. In order for FlexNet Manager for Engineering Applications to process LUM report log files, each file must contain data from only one license server and vendor.

To assist you with splitting LUM event files to ensure that each file contains events from only one license server and vendor daemon, FlexNet Manager for Engineering Applications provides a command-line utility, `processLUMFile`, to split the files. After you run the utility—specifying a LUM report log file or a folder containing multiple LUM report log files—the split report log files are put in the specified output location. The split files can be imported into FlexNet Manager for Engineering Applications to be used as the basis for LUM usage reporting.

For example, if a single LUM event file contains usage data for license servers X and Y and both of the license servers serve licenses for vendors A and B, then running the command `processLUMFile` on the file would output four files showing usage data for:

- license server X, vendor A
- license server X, vendor B
- license server Y, vendor A
- license server Y, vendor B



Task

To split LUM report log files:

1. Open a Command prompt and switch to the FlexNet Manager for Engineering Applications installation directory.
2. Run the following command:

```
processLUMFile -f<input_file_or_folder> -o<output_folder>
```

Table 6-1 • processLUMFile Parameters

| Parameter | Description |
|---|--|
| <code>-f<input_file_or_folder></code> | Specifies the LUM report file or folder of report files to be split. If the path name contains spaces, then the value must be enclosed in double quotes. |
| <code>-o<output_folder></code> | Specifies the output directory for the split files. If the path name contains spaces, then the value must be enclosed in double quotes. |

Example

```
flexnet processLUMFile -f"c:\inputDir\testLUMFile.txt" -o"c:\outputDir"
flexnet processLUMFile -f"c:\input" -o"c:\outputDir"
```

Reporting on non-FlexNet and non-LUM Application Usage

FlexNet Manager for Engineering Applications provides common report log importing—available with a separate license from Flexera—which enables enterprises to define a report log format, then import those report logs into FlexNet Manager for Engineering Applications using the FlexNet Manager for Engineering Applications interface or at the command line.

This functionality enables enterprises to import non-FlexNet (and non-LUM) report logs for usage reporting and tracking purposes. In this way, FlexNet Manager for Engineering Applications provides for comprehensive reporting across your engineering and technical application usage and enables you to do consolidated planning for your enterprise-wide software licensing needs.

The process for working with non-FlexNet report log data is:

1. Create a report definition in FlexNet Manager for Engineering Applications, using functionality available from the Report Log Definitions tab.
2. Format your report log files properly, generate or convert them to match the formats that you have defined.
3. Import the properly formatted report logs using the command-line `bulkimport` utility or using the Import button on the Report Logs screen.
4. Create and generate reports based on your usage data.

See the online FlexNet Manager for Engineering Applications 2020 R1 Help Library for detailed information.

Classic Reporting Data Model

For increased reporting efficiency, FlexNet Manager for Engineering Applications uses the FlexNet Reporting database to store usage data and generate report output. When report log data is imported into FlexNet Manager for Engineering Applications (whether on a scheduled or manual basis), that data is parsed and put into the FlexNet Reporting database.

This appendix provides information about the Classic reporting data model, including table names, column names, data types and lengths, descriptions, and any constraints.

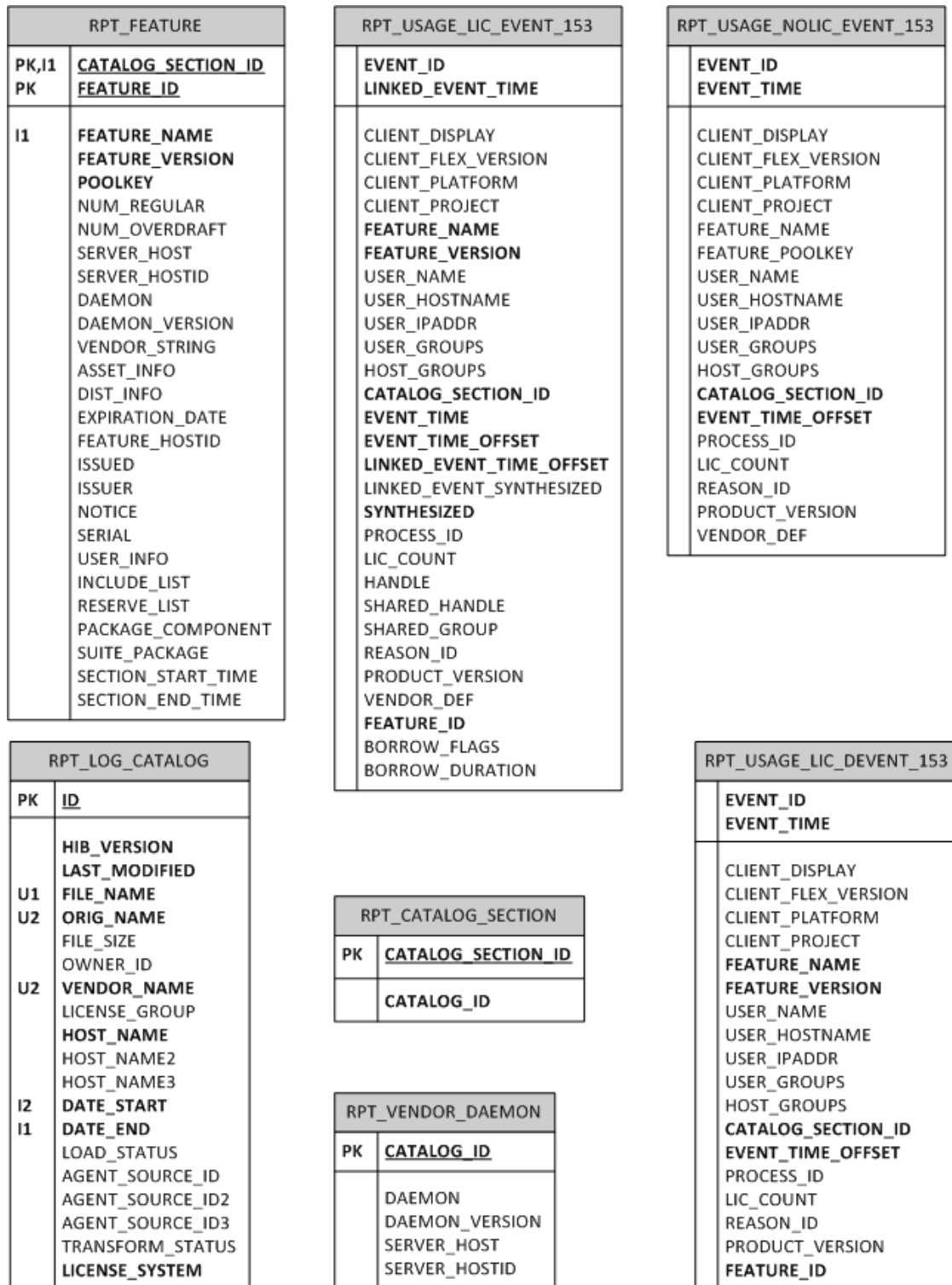


Figure 7-1: FlexNet Manager for Engineering Applications Classic Reporting schema

The CATALOG_ID columns

Every report log that is imported is assigned a unique numeric identifier by the loading process. All data in the database that is extracted from this report log is tagged with this identifier so that their source can be easily identified.

This numeric ID can be matched back to the ID of the ReportLogCatalog object (stored in the RPT_LOG_CATALOG table of the FlexNet database). The information on the ReportLogCatalog object includes the name of the report log file, its import date, and file size.

The CATALOG_SECTION_ID columns

Every report log consists of one or more sections. A section is the period between the vendor daemon's start and stop or re-reads and where the set of features is static. Typically, the set of features for a section is read at the start of the section and does not change until the section ends.

A section can start because of a vendor daemon start, a report log rotation, or a reread command executed by a vendor daemon. A section can end because of a vendor daemon shutdown, a report log rotation, a reread command executed by a vendor daemon, or a server crash. In the last case, the report log section is not ended by the vendor daemon (because it has crashed), but is synthesized by the report data loader. The CATALOG_SECTION_ID is guaranteed to be unique across the sections of all report logs.

RPT_CATALOG_SECTION Table

This table holds information about the mapping between the CATALOG_ID and CATALOG_SECTION_ID.

Table 7-1 • RPT_CATALOG_SECTION Table

| Column Name | Data Type | Data Length | Description | Constraints |
|--------------------|-----------|-------------|--|---|
| CATALOG_ID | Integer | 4 | The ID of the imported report log from where this user was extracted. | Not nullable. Part of the primary key of the table. |
| CATALOG_SECTION_ID | Integer | 4 | The ID of the section in the imported report log from where this user was extracted. | Not nullable. Part of the primary key of the table. |

RPT_FEATURE Table

This table holds information about the license features that are found in the report logs that were imported.

Table 7-2 • RPT_FEATURE Table

| Column Name | Data Type | Data Length | Description | Constraints |
|--------------------|-----------|-------------|---|---|
| CATALOG_SECTION_ID | Integer | 4 | The ID of the section in the imported report log from where this feature was extracted. | Not nullable. Part of the primary key of the table. |

Table 7-2 • RPT_FEATURE Table

| Column Name | Data Type | Data Length | Description | Constraints |
|---------------------------|------------------|-------------|---|---|
| FEATURE_ID | Integer | 4 | An ID assigned to the feature record in the table. It is unique across all the feature records in a section of a report log; hence the two IDs, CATALOG_SECTION_ID and FEATURE_ID form the unique key for RPT_FEATURE Table . | Not nullable. Part of the primary key of the table. |
| FEATURE_NAME | String | 255 | The name of the FEATURE. | Not nullable. |
| FEATURE_VERSION | String | 100 | The version of the FEATURE or INCREMENT line in the license files of the vendor daemon. | Not nullable. |
| POOLKEY | String | 255 | The license key on the FEATURE or INCREMENT line in the license file of the vendor daemon. | Not nullable. |
| SECTION_START_TIME | Integer | 4 | Section start time identifies the moment a feature becomes available at the beginning of a section. The date is in seconds from Jan 1, 1970. | |
| SECTION_END_TIME | Integer | 4 | Section end time identifies the moment of time when a feature becomes unavailable at the end of the section. The date is in seconds from Jan 1, 1970. | |
| NUM_REGULAR | Nullable Integer | 2 | The number of regular licenses for this feature. A NULL value in this property denotes uncounted licenses. | |
| NUM_OVERDRAFT | Nullable Integer | 2 | The number of overdraft licenses for this feature. | |
| SERVER_HOST | Nullable String | 200 | The name of the server that hosted the vendor daemon that processed the event. | |
| SERVER_HOSTID | Nullable String | 200 | The hostid of the server that hosted the vendor daemon that processed the event. | |
| DAEMON | Nullable String | 200 | The name of the vendor daemon that processed the event. | |
| DAEMON_VERSION | Nullable String | 100 | The version of the vendor daemon that processed the event. | |
| VENDOR_STRING | Nullable String | 255 | The text value of the VENDOR_STRING parameter on the FEATURE or INCREMENT line in the license file of the vendor daemon. | |
| ASSET_INFO | Nullable String | 255 | The text data that is provided with the asset_info parameter on the FEATURE or INCREMENT line in the license file of the vendor daemon. | |

Table 7-2 • RPT_FEATURE Table

| Column Name | Data Type | Data Length | Description | Constraints |
|------------------------|------------------|-------------|---|-------------|
| DIST_INFO | Nullable String | 255 | The text data that is provided with the <code>dist_info</code> parameter on the <code>FEATURE</code> or <code>INCREMENT</code> line in the license file of the vendor daemon. | |
| EXPIRATION_DATE | Integer | 8 | The expiration date of the <code>FEATURE</code> or <code>INCREMENT</code> line in the license file of the vendor daemon. The date is given in seconds from Jan 1, 1970. A <code>NULL</code> value in this property denotes a permanent license. | |
| FEATURE_HOSTID | Nullable String | 200 | The text data for <code>HOSTID</code> found on the <code>FEATURE</code> or <code>INCREMENT</code> line in the license file of the vendor daemon. | |
| ISSUED | Nullable Integer | 4 | The date on which the feature was issued, as defined by the <code>ISSUED</code> parameter on the <code>FEATURE</code> or <code>INCREMENT</code> line in the license file of the vendor daemon. The date is in seconds from Jan 1, 1970. | |
| ISSUER | Nullable String | 200 | The text data that is the issuer of the feature line, as defined by the <code>ISSUER</code> parameter on the <code>FEATURE</code> or <code>INCREMENT</code> line in the license file of the vendor daemon. | |
| NOTICE | Nullable String | 255 | The text data that is the notice value, as defined by the <code>NOTICE</code> parameter on the <code>FEATURE</code> or <code>INCREMENT</code> line in the license file of the vendor daemon. | |
| SERIAL | Nullable String | 200 | The text data that is the serial number that is defined by the <code>SN</code> parameter on the <code>FEATURE</code> or <code>INCREMENT</code> line in the license file of the vendor daemon. | |
| USER_INFO | Nullable String | 255 | The text data that is the user information defined by the <code>user_info</code> parameter on the <code>FEATURE</code> or <code>INCREMENT</code> line in the license file of the vendor daemon. | |
| INCLUDE_LIST | Nullable String | 1024 | A string that concatenates the names—in alphabetical order—of all the users and user groups that have been <code>INCLUDED</code> for this feature as defined in the options file. The feature's license key is the only characteristic that is used to identify the feature. The user and user group names are surrounded by double quotes and separated by commas. | |

Table 7-2 • RPT_FEATURE Table

| Column Name | Data Type | Data Length | Description | Constraints |
|--------------------------|-----------------|-------------|--|---------------------------|
| RESERVE_LIST | Nullable String | 1024 | <p>A string that concatenates the names—in alphabetical order—of all the users and user groups that have been RESERVED for this feature as defined in the options file.</p> <p>The feature's license key is the only characteristic that is used to identify the feature.</p> <p>The user and user group names are surrounded by double quotes and separated by commas.</p> | |
| PACKAGE_COMPONENT | Nullable String | 1 | <p>A value of "Y" denotes that the feature is a component of a package. Only the component of a package is flagged as "Y"; the package feature itself is not.</p> <p>A value of "N" denotes that the feature is not a component of a package. In this case, the feature can be either the package itself or not associated with any package.</p> | Must be either "Y" or "N" |
| SUITE_PACKAGE | Nullable String | 1 | <p>A value of "Y" denotes that the feature is a suite package. Only the package feature of the suite package is flagged with a "Y". The components of the suite package are not.</p> <p>A value of "N" denotes that the feature is not a suite package. In this case, the feature can be either the component of the suite package itself or a package or component that is not associated with any suite package whatsoever or a feature that is not associated with any package.</p> | Must be either "Y" or "N" |

RPT_VENDOR_DAEMON Table

This table holds information about a vendor daemon instance that a report log file identified by the CATALOG_ID came from.

Table 7-3 • RPT_VENDOR_DAEMON Table.

| Column Name | Data Type | Data Length | Description | Constraints |
|-------------------|-----------|-------------|---|--|
| CATALOG_ID | Integer | 4 | The ID of the section in the imported report log from where this event was extracted. | Not nullable. Part of the table's primary key. |

| Column Name | Data Type | Data Length | Description | Constraints |
|-----------------------|-----------------|-------------|--|-------------|
| DAEMON | Nullable String | 200 | The name of the vendor daemon that provided a report log file identified by the CATALOG_ID. | |
| DAEMON_VERSION | Nullable String | 100 | The version of the vendor daemon that provided a report log file identified by the CATALOG_ID. | |
| SERVER_HOST | Nullable String | 200 | The name of the server that hosted the daemon instance that provided a report log file identified by the CATALOG_ID. | |
| SERVER_HOSTID | Nullable String | 200 | The HOSTID of the server that hosted the daemon instance that provided a report log file identified by the CATALOG_ID. | |

RPT_ERROR Table

This table holds all the errors that were detected during the report log import process.

(This table is not shown in the [reporting schema diagram](#).)

Table 7-4 • RPT_ERROR Table

| Column Name | Data Type | Data Length | Description | Constraints |
|-------------------|-----------|-------------|--|---------------|
| CATALOG_ID | Integer | 4 | The ID of the imported report log in which the error was detected. | Not nullable. |
| ERROR_TIME | Integer | 4 | The time that the error occurred in the report log. | Not nullable. |

Table 7-4 • RPT_ERROR Table

| Column Name | Data Type | Data Length | Description | Constraints |
|--------------------|-----------|-------------|--|---------------|
| ERROR_CODE | Integer | 2 | <p>A numeric code that denotes the error that was detected. You can use this value to selectively process records from this table.</p> <ul style="list-style-type: none"> 1001 denotes that the report log section did not have an END event, probably due to log corruption. 1002 denotes that the CONFIG line for a license line referenced by a feature list line could not be found. 1003 denotes that data for multiple vendor daemons was found in a single report log. 1004 denotes that the timestamps in a report log have been rolled back. 1005 denotes that no matching check-out event could be found for a check-in event in the report log. 1006 denotes that data for multiple host ids was found in a single report log. 1007 denotes that outstanding checkouts were not checked in by the end of the section. 1008 denotes that a denial event had no feature associated with it. | Not nullable. |
| SEVERITY | Integer | 1 | <p>This numeric code denotes the severity of the error as determined by the report data loader.</p> <ul style="list-style-type: none"> 0 (zero) denotes a low severity level, usually associated with a warning. The report data loader would attempt to handle the error in some manner by, for example, recovering the data as much as possible, and then the loader would proceed to process the rest of the report log. 1 denotes a fatal error. Typically, the report data loader would not proceed with the processing of the report log after a fatal error is detected and reported. | Not nullable. |
| DESCRIPTION | String | 255 | <p>A text string describing the nature of the error that occurred. This string is not localized or localizable. It contains context-sensitive information about the error.</p> | Not nullable. |

RPT_REASON Table

The RPT_REASON table holds the key-value pairs of reason IDs to their text strings describing the reasons for denial or check-in. This is a look-up table; the data in this table is never altered (inserted, edited, or deleted).

(This table is not shown in the [reporting schema diagram](#).)

Table 7-5 • RPT_REASON Table

| Column Name | Data Type | Data Length | Description | Constraints |
|-------------|-----------------|-------------|--|---|
| REASON_ID | Integer | 1 | The ID that is the key to the reason for denial or check-in. It is cross-referenced with denial and check-in events from the RPT_USAGE_NOLIC_EVENT_153 Table . | Not nullable. Part of the primary key of the table. |
| REASON | Nullable String | 255 | The text strings describing the reasons for denials and check-ins. | |

RPT_USAGE_LIC_EVENT_153 Table

This table holds information about the client processes that executed events found in the report logs that were imported.

Table 7-6 • RPT_USAGE_LIC_EVENT_153 Table

| Column Name | Data Type | Data Length | Description | Constraints |
|---------------------|-----------------|-------------|---|---------------|
| CLIENT_DISPLAY | Nullable String | 255 | The user's display name (UNIX only). | |
| CLIENT_FLEX_VERSION | Nullable String | 255 | The product's FlexNet Licensing version. | |
| CLIENT_PLATFORM | Nullable String | 255 | The name of the platform on which the user's application was compiled to run. | |
| CLIENT_PROJECT | Nullable String | 255 | The user's <i>LM_PROJECT</i> environment variable (or the related value in the Windows registry). | |
| FEATURE_NAME | String | 255 | The name of the FEATURE. | Not nullable. |
| FEATURE_VERSION | String | 100 | The version of the FEATURE or INCREMENT line in the license files of the vendor daemon. | Not nullable. |
| USER_NAME | Nullable String | 255 | The user's user name (from report log files), in uppercase. If the user name is in mixed case or lowercase, FlexNet Manager for Engineering Applications converts the string to uppercase before storing it in the reporting database. | |

Table 7-6 • RPT_USAGE_LIC_EVENT_153 Table

| Column Name | Data Type | Data Length | Description | Constraints |
|---------------------------|-----------------|-------------|--|---|
| USER_HOSTNAME | Nullable String | 255 | The user's host name. | |
| USER_IPADDR | Nullable String | 255 | The user's IP address in string form. | |
| USER_GROUPS | Nullable String | 1024 | <p>A string that is the concatenation of the names of all the user groups to which the user belongs. The user's group membership is defined in the options file.</p> <p>The user group names are surrounded by double quotes and separated by commas.</p> | |
| HOST_GROUPS | Nullable String | 1024 | <p>A string that is the concatenation of the names of all the host groups to which the user's host belongs. The user's host's group membership is defined in the options file.</p> <p>The host group names are surrounded by double quotes and separated by commas.</p> | |
| CATALOG_SECTION_ID | Integer | 4 | The ID of the section in the imported report log from where this feature was extracted. | Not nullable. Part of the primary key of the table. |
| EVENT_TIME | Integer | 4 | The start time of the event. The time is given in seconds from Jan 1, 1970. | Not nullable. Part of the primary key of the table. |
| EVENT_TIME_OFFSET | Integer | 2 | <p>A running count of the number of events in a section of a report log registered for the same second.</p> <p>This provides a means to sequence the events as they were found in the report log. If multiple events occurred at the same point in time, they are assigned unique offset numbers depending on their order in the report log.</p> <p>By sorting on this column, you can see events coming from the same report log section, in the order they were logged by the vendor daemon.</p> | Not nullable. Part of the primary key of the table. |

Table 7-6 • RPT_USAGE_LIC_EVENT_153 Table

| Column Name | Data Type | Data Length | Description | Constraints |
|---------------------------------|-----------|-------------|---|---------------|
| LINKED_EVENT_TIME_OFFSET | Integer | 2 | <p>The linked event time offset of the event. For more details, see the description for EVENT_TIME_OFFSET property. The time is given in seconds from Jan 1, 1970.</p> <p>See the note in the LINKED_EVENT_TIME description for information about how CHECK_OUT and QUEUED events are treated.</p> | |
| LINKED_EVENT_SYNTHESIZED | Integer | 1 | <p>An ID denoting whether the linked event was an actual event or if it was synthesized. For more details, see the description for SYNTHESIZED.</p> <p>See the note in the LINKED_EVENT_TIME description for information about how CHECK_OUT and QUEUED events are treated.</p> | |
| SYNTHESIZED | Integer | 1 | <p>An ID indicating whether this event was an actual event or if it was synthesized.</p> <p>Some events do not actually take place, but are synthesized by the vendor daemon. For example, when a section is ended by the vendor daemon, all features that are still in a checked-out state are forcibly checked in by the vendor daemon by synthesizing a check-in event for each outstanding check-out.</p> <p>In addition, when a section starts because of a report log rotation or a reread of the license files, all outstanding check-outs that were forcibly checked in at the end of the previous section are forcibly checked out again by the vendor daemon by synthesizing checkout events for each synthesized check-in at the previous section end. Thus, when a report log is rotated or a reread is performed, there may be several pairs of check-ins and check-outs that are synthesized at the section end and start, respectively. These events did not actually occur; however, by synthesizing them, the vendor daemon guarantees that every section contains a pair of check-out and check-in events that match.</p> | Not nullable. |
| PROCESS_ID | Integer | 8 | The ID of the user's process that initiated the event. | |

Table 7-6 • RPT_USAGE_LIC_EVENT_153 Table

| Column Name | Data Type | Data Length | Description | Constraints |
|------------------------|------------------|-------------|---|---|
| LIC_COUNT | Nullable Integer | 8 | The number of licenses in request. This column is NULL for START and END events. | |
| HANDLE | Nullable Integer | 4 | An identifier assigned by a vendor daemon to a license check-out event. | |
| SHARED_HANDLE | Nullable Integer | 4 | An identifier assigned by the vendor daemon for a previous license checkout with which this checkout was shared. | |
| SHARED_GROUP | Nullable Integer | 4 | An identifier assigned by a vendor daemon, representing a group of one or more license checkouts which were satisfied with the same license. | |
| REASON_ID | Nullable Integer | 1 | An identifier referencing the reason a license was denied or checked in. | |
| PRODUCT_VERSION | Nullable String | 255 | The version of the product used by the user. | |
| VENDOR_DEF | Nullable String | 255 | The value of the CHECKOUT_DATA passed by the user's application to the license server when the license was checked out. | |
| FEATURE_ID | Integer | 4 | An ID assigned to the feature record in the table. It is unique across all the feature records in a section of a report log; hence the two IDs, CATALOG_SECTION_ID and FEATURE_ID form the unique key for RPT_FEATURE Table . | Not nullable. Part of the primary key of the table. |
| BORROW_FLAGS | Nullable Integer | 2 | BORROW indicator from FlexNet Publisher. | |
| BORROW_DURATION | Nullable Number | 19 | Time (in seconds) during which the license is being borrowed. | |

RPT_USAGE_NOLIC_EVENT_153 Table

This table holds information about the client processes that executed events found in the report logs that were imported.

Table 7-7 • RPT_USAGE_NOLIC_EVENT_153 Table

| Column Name | Data Type | Data Length | Description | Constraints |
|----------------------------|-----------------|-------------|---|---|
| EVENT_ID | Integer | 1 | An ID denoting the event type. The values are: <ul style="list-style-type: none"> 4 for a CHECK-IN event 5 for an ordinary DENIAL event 10 for a UNSUPPORTED event 12 for a ULTIMATE DENIAL event 14 for a DEQUEUED event | Not nullable. Part of the primary key of the table. |
| EVENT_TIME | Integer | 4 | The start time of the event. The time is given in seconds from Jan 1, 1970. | Not nullable. Part of the primary key of the table. |
| CLIENT_DISPLAY | Nullable String | 255 | The user's display name (UNIX only). | |
| CLIENT_FLEX_VERSION | Nullable String | 255 | The product's FlexNet Licensing version. | |
| CLIENT_PLATFORM | Nullable String | 255 | The name of the platform on which the user's application was compiled to run. | |
| CLIENT_PROJECT | Nullable String | 255 | The user's <i>LM_PROJECT</i> environment variable (or the related value in the Windows registry). | |
| FEATURE_NAME | String | 255 | The name of the FEATURE. | Not nullable. |
| FEATURE_POOLKEY | Nullable String | 255 | The poolkey of the feature (if one is associated with this event). The poolkey of the license feature is normally taken from the RPT_FEATURE Table using the feature referenced by the <i>FEATURE_ID</i> column. In the case of UNSUPPORTED events, however, the feature is not available in the RPT_FEATURE Table , so this column in the <i>RPT_EVENT</i> table is used as an alternate source for the feature poolkey. | |
| USER_NAME | Nullable String | 255 | The user's user name (from report log files), in uppercase. If the user name is in mixed case or lowercase, FlexNet Manager for Engineering Applications converts the string to uppercase before storing it in the reporting database. | |

Table 7-7 • RPT_USAGE_NOLIC_EVENT_153 Table

| Column Name | Data Type | Data Length | Description | Constraints |
|---------------------------|------------------|-------------|--|---|
| USER_HOSTNAME | Nullable String | 255 | The user's host name. | |
| USER_IPADDR | Nullable String | 255 | The user's IP address in string form. | |
| USER_GROUPS | Nullable String | 1024 | <p>A string that is the concatenation of the names of all the user groups to which the user belongs. The user's group membership is defined in the options file.</p> <p>The user group names are surrounded by double quotes and separated by commas.</p> | |
| HOST_GROUPS | Nullable String | 1024 | <p>A string that is the concatenation of the names of all the host groups to which the user's host belongs. The user's host's group membership is defined in the options file.</p> <p>The host group names are surrounded by double quotes and separated by commas.</p> | |
| CATALOG_SECTION_ID | Integer | 4 | The ID of the section in the imported report log from where this feature was extracted. | Not nullable. Part of the primary key of the table. |
| EVENT_TIME_OFFSET | Integer | 2 | <p>A running count of the number of events in a section of a report log registered for the same second.</p> <p>This provides a means to sequence the events as they were found in the report log. If multiple events occurred at the same point in time, they are assigned unique offset numbers depending on their order in the report log.</p> <p>By sorting on this column, you can see events coming from the same report log section, in the order they were logged by the vendor daemon.</p> | Not nullable. Part of the primary key of the table. |
| PROCESS_ID | Integer | 8 | The ID of the user's process that initiated the event. | |
| LIC_COUNT | Nullable Integer | 2 | <p>The number of licenses in request.</p> <p>This column is NULL for START and END events.</p> | |
| REASON_ID | Nullable Integer | 1 | An identifier referencing the reason a license was denied or checked in. | |

Table 7-7 • RPT_USAGE_NOLIC_EVENT_153 Table

| Column Name | Data Type | Data Length | Description | Constraints |
|------------------------|-----------------|-------------|---|-------------|
| PRODUCT_VERSION | Nullable String | 255 | The version of the product used by the user. | |
| VENDOR_DEF | Nullable String | 255 | The value of the CHECKOUT_DATA passed by the user's application to the license server when the license was checked out. | |

RPT_USAGE_LIC_DEVENT_153 Table

This table holds information about the client processes that executed events found in the report logs that were imported.

Table 7-8 • RPT_USAGE_LIC_DEVENT_153 Table

| Column Name | Data Type | Data Length | Description | Constraints |
|----------------------------|-----------------|-------------|--|---|
| EVENT_ID | Integer | 1 | An ID denoting the event type. The values are: <ul style="list-style-type: none"> 4 for a CHECK-IN event 5 for an ordinary DENIAL event 10 for a UNSUPPORTED event 12 for a ULTIMATE DENIAL event 14 for a DEQUEUED event | Not nullable. Part of the primary key of the table. |
| EVENT_TIME | Integer | 4 | The start time of the event. The time is given in seconds from Jan 1, 1970. | Not nullable. Part of the primary key of the table. |
| CLIENT_DISPLAY | Nullable String | 255 | The user's display name (UNIX only). | |
| CLIENT_FLEX_VERSION | Nullable String | 255 | The product's FlexNet Licensing version. | |
| CLIENT_PLATFORM | Nullable String | 255 | The name of the platform on which the user's application was compiled to run. | |
| CLIENT_PROJECT | Nullable String | 255 | The user's <i>LM_PROJECT</i> environment variable (or the related value in the Windows registry). | |
| FEATURE_NAME | String | 255 | The name of the FEATURE. | Not nullable. |
| FEATURE_VERSION | String | 100 | The version of the FEATURE or INCREMENT line in the license files of the vendor daemon. | Not nullable. |

Table 7-8 • RPT_USAGE_LIC_DEVENT_153 Table

| Column Name | Data Type | Data Length | Description | Constraints |
|---------------------------|-----------------|-------------|--|---|
| USER_NAME | Nullable String | 255 | <p>The user's user name (from report log files), in uppercase.</p> <p>If the user name is in mixed case or lowercase, FlexNet Manager for Engineering Applications converts the string to uppercase before storing it in the reporting database.</p> | |
| USER_HOSTNAME | Nullable String | 255 | The user's host name. | |
| USER_IPADDR | Nullable String | 255 | The user's IP address in string form. | |
| USER_GROUPS | Nullable String | 1024 | <p>A string that is the concatenation of the names of all the user groups to which the user belongs. The user's group membership is defined in the options file.</p> <p>The user group names are surrounded by double quotes and separated by commas.</p> | |
| HOST_GROUPS | Nullable String | 1024 | <p>A string that is the concatenation of the names of all the host groups to which the user's host belongs. The user's host's group membership is defined in the options file.</p> <p>The host group names are surrounded by double quotes and separated by commas.</p> | |
| CATALOG_SECTION_ID | Integer | 4 | The ID of the section in the imported report log from where this feature was extracted. | Not nullable. Part of the primary key of the table. |
| EVENT_TIME_OFFSET | Integer | 2 | <p>A running count of the number of events in a section of a report log registered for the same second.</p> <p>This provides a means to sequence the events as they were found in the report log. If multiple events occurred at the same point in time, they are assigned unique offset numbers depending on their order in the report log.</p> <p>By sorting on this column, you can see events coming from the same report log section, in the order they were logged by the vendor daemon.</p> | Not nullable. Part of the primary key of the table. |
| PROCESS_ID | Integer | 8 | The ID of the user's process that initiated the event. | |

Table 7-8 • RPT_USAGE_LIC_DEVENT_153 Table

| Column Name | Data Type | Data Length | Description | Constraints |
|------------------------|------------------|-------------|---|---|
| LIC_COUNT | Nullable Integer | 8 | The number of licenses in request. This column is NULL for START and END events. | |
| REASON_ID | Nullable Integer | 1 | An identifier referencing the reason a license was denied or checked in. | |
| PRODUCT_VERSION | Nullable String | 255 | The version of the product used by the user. | |
| FEATURE_ID | Integer | 4 | An ID assigned to the feature record in the table. It is unique across all the feature records in a section of a report log; hence the two IDs, CATALOG_SECTION_ID and FEATURE_ID form the unique key for RPT_FEATURE Table . | Not nullable. Part of the primary key of the table. |

Generating Command-Line Reports

Available only for FlexNet Manager for Engineering Applications's Classic reports, command-line reporting functionality in FlexNet Manager for Engineering Applications is provided to support legacy back-office systems of customers migrating from SAMSuite to FlexNet Manager for Engineering Applications. All functionality provided by command-line reporting can be achieved using the FlexNet Manager for Engineering Applications interface.

You can configure an entire report on the command line, without the use of configuration files. For more information, run the command `flexnet report -help` in the FlexNet Manager for Engineering Applications installation directory.

The report log must be imported into FlexNet Manager for Engineering Applications before a report can be created. Use the `bulkimport` function to import the log and then create a report.

[Table 8-1](#) offers a starting point for each type of report. To construct a batch report configuration entirely on the command line, use the appropriate basic command, adding input, output, format, and other applicable switches.

Table 8-1 • Summary of Commands to Invoke Report Types and Output Formats

| Report Family | Report Type | Basic Commands | Output Formats |
|---------------|-----------------------|---|------------------------------|
| Summary | Usage Over Time | report Summary -uot -interval period | RIF JPEG/HTML PNG/HTML |
| | High Water Mark | report Summary - hwmot -interval period | RIF JPEG/HTML PNG/HTML |
| | Summary Barchart | report Summary -barchart metric | JPEG/HTML PNG/HTML |
| | Usage Efficiency | report Summary- usage_efficiency -barchart metric | RIF JPEG/HTML PNG/HTML |
| | Usage Summary | report Summary -usage_summary | HTML Text RIF |
| | Peak Usage and Denial | report Summary -pud | JPEG/HTML PNG/HTML |
| | Unique User | report Summary -uu | HTML, XML |
| Server | Server Coverage | report Server -coverage | Text |
| Raw | Raw | report Raw | Text RIF, XML |

Running Reports from the Command Line

You can run a report from the command line, by running it from the directory in which FlexNet Manager for Engineering Applications Admin is installed.

**Task****To run a report from the command line:**

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

```
flexnet report (Raw|Summary|Server) -user <user> -password <password> -host <host>
      -output_dir <output_dir> -output <report_name>
      (-input <files> | -input_single <single_file>)
      [OPTIONS]
```

Even if the configuration file contains the report family (for example, Summary) the report family must be specified as part of the command. parameters can be specified in any order.

Example

```
flexnet report Summary -host http://localhost:8888 -user admin -password admin -input test5.r1
      -output Usage_Summary -output_dir c:\Test -rif -usage_summary -title Usage_Summary_Title
      -startdate "12/21/2006_18:05:12" -enddate "12/21/2006_18:10" -html -breakdown user
      -filter feature "==" "gz1" -filterperiod 1.17:56:34 7.19:11:23 -config config1.config
```

Defining Input and Output Files

You can generate a single report from one or more FlexNet Licensing report logs, written by one or more vendor daemons, as long as each report log is written by a single vendor daemon running on a single host (vendor daemons running on more than one server must each write to their own report logs). The report log files must be locally accessible (via NFS) to FlexNet Manager for Engineering Applications Reporting. If you want to use more than one report log, enter the path to each report log separately. If you use **-input** (which is optional in FlexNet Manager for Engineering Applications, as the report logs have already been loaded to the database), these paths must not contain spaces.

```
-input input_filename1 [input_filename2 ...]
```

However, if you have one or more input files (on Windows) whose names contain spaces, specify those files one at a time with **-input_single** rather than **-input** and surround the file name with double quotes. For example:

```
-input_single "file name 1" -input_single "file name 2"
```

Writing an output file is mandatory for reports with text, HTML, RIF, or JPEG/HTML output. To specify output, you must provide the output format(s) and an output file name base (including path). The paths must not contain spaces.

```
-output_format1 -output_format2 ... -output output_filename_base
```

The same file name base is used for all of the output files written during the generation of a report. The file name extension is added automatically, depending on the report format (.html, .txt, .rif, or .jpg/.htm).

The output formats available from the command line are listed below:

Table 8-2 • Output Formats and Their Command-Line Parameters

| Output Format | Command-Line Parameter |
|---------------|------------------------|
| HTML file | -html |

Table 8-2 • Output Formats and Their Command-Line Parameters

| Output Format | Command-Line Parameter |
|---------------------------------|------------------------|
| Text file | -text |
| Report Interchange Format files | -rif |
| JPEG/HTML files | -jpeg |
| PNG/HTML files | -png |
| XML | -xml |

Defining the Level of Detail in a Report

All reports show usage by feature. To create a report that shows more detail about the nature of license usage, use one or more of the following settings:

- Summary period
- Breakdown category
- Sort order specification
- Feature description (distinguishing different license pools)

A summary period is mandatory for Usage Over Time and High Water Mark reports. The default value for a Usage Over Time report is None (one second) and for a High Water Mark report is Hour.

A different summary period is specified by typing:

(-uot) -interval period

A summary period for Summary Barchart and Usage Summary reports specifies that the report period should be decomposed into the requested time intervals. A summary period is specified by typing:

-periodic -interval period

Available summary periods are None, Minute, Hour, Day, Week, Semi_Month, and Month.

A breakdown category specifies that the license usage be characterized by the specified attribute. Breakdown categories are specified by typing:

-breakdown category1 [-breakdown category2 ...]

The sort order of report output is specified by typing:

-orderby item(s)

You can order report output by feature (f), breakdown category (b), first by feature, then by breakdown category (fb), or first by breakdown category, then by feature (bf).

You can distinguish the usage of different license pools of a feature by typing:

-feat_uniq id

where id is {name name_ver name_key name_asset}.

Defining Field Filters

At the command line, specify an event filter condition using:

`-filter category "operator" value`

For example:

`-filter feature "=~" toolbox`

where:

Table 8-3 • Defining Field Filters

| | |
|----------|-----------------------|
| category | Filter category. |
| operator | Relational operator. |
| value | Value for comparison. |

The following rules for filtering of data are applied:

Table 8-4 • Rules for filtering of data

| Input filename specified? | Date range specified? | Data reported |
|---------------------------|-----------------------|--|
| no | no | everything (no filters) |
| no | yes | everything in the date range |
| yes | no | everything in the files |
| yes | no | data in files for the specified date range |

FlexNet Manager for Engineering Applications supports values with spaces. You will need to double-quote the values. You do not need to substitute asterisks for spaces.

Table 8-5 lists valid operators in FlexNet Manager for Engineering Applications Reporting filters.



Important • All special characters must be enclosed in quotes (“==”). For example, all paths with spaces, all filtering operators (== != =~ ><), all dates with commas and spaces, and all delimiters, should be surrounded with quotes.

Table 8-5 • Valid Operators in FlexNet Manager for Engineering Applications Reporting Filters

| Operator | Description |
|----------|--|
| == | Identical to |
| ==~ | Includes (with wildcards) <ul style="list-style-type: none"> * Match any number of characters. ? Match any single character. Asterisks are not required at the front or at the end of the operator, so if you have the filter user ==~john, you could get results including “aajohn”, “johnbb” and “john”. |
| != | Not equal to |
| !~ | Excludes (with wildcards) See discussion for ==~. |
| > | Greater than |
| < | Less than |

Time period filters set time periods during each week to include in a report. One or more time period filters can be set by specifying:

-filterperiod start end

where start and end time formats are [d.]HH:mm[:ss] and d is day 1-7 of the week (Sun-Sat).

You can select whether to include in your report output features that have no license usage by typing:

-showzero

You can select whether to eliminate “false” denials in your report output by typing:

-elim_false_denials [interval]

where interval is the time period in seconds in which to consider a series of “false” denials for elimination. If not specified, the default interval is 10 seconds.

You can select which event types to include in your report by typing:

-event_type type

where type is {used denied queued exception checkout}.

If no specific types are specified, all event types except Checkout are included in a report.

You can select dates to start and end the data in your report by typing:

-startdate date

-enddate date

At the command line, quotation marks around date strings containing spaces sometimes cause problems on particular operating systems. Windows users must always include quotation marks around a date string. On UNIX, FlexNet Manager for Engineering Applications Reporting accepts an underscore instead of a space in a date string, which makes quotation marks unnecessary. For example, "03/03/1999 02:00" is equivalent to 03/03/1999_02:00.

Customizing Appearance of Graphical Report Output

The default graphical configuration is managed by templates, and is the same as the graphical configuration for the reports run through the GUI. You can use the command-line options to request a specific predefined template. See [Command-Line Reference](#).

There are several options that allow you to customize the text style and colors used in graphical report output. Colors should be specified in hexadecimal RGB format (for example, 00AAFF). Do not use plain words; however, the colors from SAMreport are accepted. SAMreport colors are mapped as shown in the following table.

Table 8-6 • SAMreport Color Mapping

| SAMreport Color | Hexadecimal Format |
|-----------------|--------------------|
| black | 000000 |
| red | FE0000 |
| green | 008200 |
| blue | 0000FE |
| cyan | 00FEFE |
| magenta | FE00FE |
| yellow | FEFE00 |
| white | FFFFFF |

The default title of a graphical report depends on the type of report. You can set the title of a graph, but note that all individual output graphs in a set will have the same title. If you want to set the title to something other than the default report title, type:

-title title

Background color of the graphical report can be set by typing:

`-bgcolor color`

Several font characteristics can be set for all text in a graphical report. Some options require that additional options are set.

Table 8-7 • Font Settings for Graphical Reports

| Option | Description | Constraints |
|---|--|--|
| <code>-fontface <fontname></code> | Sets the name of the font to use for text in the report. Specify one of the fonts accessible by the Java VM. | If you set one of these options, you must also set the other. |
| <code>-fontsize <points></code> | Sets the size of the font in points for text in the report. | |
| <code>-fontbold</code> | Make text bold in the report | If set, must also set <code>-fontface</code> , <code>-fontsize</code> . |
| <code>-fontitalic</code> | Make text italicized in the report. | If set, must also set <code>-fontface</code> , <code>-fontsize</code> . |

The color of all text in a graphical report can be set by typing:

`-textcolor color`

The color(s) of data elements are specified in an order dependent on the type of report. In a Usage Over Time or a High Water Mark report, the first color is used for the line indicating usage; the second color is used for the line indicating available licenses. In a barchart, the specified color(s) are assigned to bars in the order they appear in the report. One or more colors are specified by typing:

`-datacolor color[,color]*`

The style of data lines can be set for one or both data lines in a graphical report. The first style is used for the line indicating usage; the second style is used for the line indicating available licenses.

`-datastyle n[,n]*`

Table 8-8 • Style of Data Lines

| | |
|----|---|
| -1 | Solid line. If <code>datastyle</code> is not set at all, the lines will be solid. |
| 0 | Dashed line. |
| 1 | Dotted line. |
| 2 | Dotted-dashed line. |

The width of all lines in the graph (data lines and axes) can be set by typing:

`-linewidth pixels`

Using a Configuration File

When generating a report from the command line, FlexNet Manager for Engineering Applications Reporting obtains report configuration information from the command line explicitly, from a plain-text configuration file (available from SAMreport, or manually entered into a text file), or both.

Command-line parameters read directly from the configuration file are indistinguishable from parameters found on the command line, with the exception of specifying a feature name. (See the following section, [Specifying Feature Names in Configuration Files](#), for information.) If conflicting values for an parameters are specified, the last value specified is applied.

Configuration files can be nested, and multiple configuration files can be specified on the command line. The paths to configuration files must not contain spaces.

Specify a configuration file by typing the following command:

```
-config config_filename
```

See Also

[Specifying Feature Names in Configuration Files](#)

Specifying Feature Names in Configuration Files

You can use a configuration file to specify report-configuration information. (See the previous section, [Using a Configuration File](#), for more information.) Within a configuration file, the way in which you specify a feature name differs from how you specify a feature name at the command line.

- If the feature name does not contain spaces, do not enclose the value or the operator (==) in quotes. For example:

```
-filter feature == feature_name
```

- If the feature name contains spaces, you must enclose the value in quotes. For example:

```
-filter feature == "feature name"
```

See Also

[Using a Configuration File](#)

Defining a Raw Report

When you generate a Raw report from the command line, you can specify which fields to display. See the section, [Raw Report Fields](#), for a list of available fields. Fields are displayed in the order specified. Each field specification consists of the `-field` parameter, the field name, the display width of the field, and the field justification:

```
-field "field_name [,col_width] [,col_justification]"
```

For example:

```
-field "event, 10, left"
```

You also have control over the length and justification of those fields. The width is specified by number of spaces and can be set according to the size of the data and number of fields in your report. Report columns can be right or left justified. By default, each entry is left-justified in its column.

These field settings produce the Raw report output that follows them:

```
-field "event,6,left"  
-field "user,8,left"  
-field "user_hostname,10"  
-field "user_display,15"  
-field "count,1"  
-field "event_time,18"  
-field "duration,11,right"
```

Table 8-9 • Raw report output

| Event | User | Host | Display | # | Time | Duration |
|--------|-------|---------|-------------|---|----------------------|-----------------|
| Unsupp | brian | ss4 | indigo2:0:0 | 1 | May 9 14:12:41 | 00d 00:00:00 |
| Used | brian | indigo2 | /dev/ttyq0 | 1 | 1 May 09 14:05:14 | ood 00:07:37 |

You can specify `-field why` to display why each license request was denied.

Command-Line Reference

This section provides information about the available command-line parameters, breakdown and filter categories, and Raw report fields.

Reporting Command-Line Parameters

[Table 8-10](#) shows the command-line parameters available for the report command. Parameters that must be set are marked as **Required**.

Command-line reporting does not check for conflicting parameters; the last parameter takes precedence. For example, if your report has both `-out` `-usage_summary` parameters, the resulting report will be Usage Summary.

The syntax for running a report from the command line (from the FlexNet Manager for Engineering Applications Admin directory) is:

flexnet report

Table 8-10 • FlexNet Manager for Engineering Applications Reporting Command-Line Parameters

| Parameter | Description |
|--------------------------------|--|
| -help | Help for the report command that lists command-line parameter usage. |
| -barchart <metric> | Specifies barchart output for Summary Barchart and Usage Efficiency reports. <metric> values are num_used, %_used, and hrs_used, max_used. |
| -bgcolor <color> | Set the background color of a graphical report. Select from: white, black, red, green, blue, cyan, magenta, yellow. The default color is white. You can also specify colors in hexadecimal RGB format. See Customizing Appearance of Graphical Report Output |
| -breakdown <category> | Specify a breakdown category . |
| -config <config_filename> | Read report parameters from a report configuration file. The file must have the file extension .config. |
| -datacolor <color[,color]> | Set the colors of data lines or bars in a graphical report. Select from: white, black, red, green, blue, cyan, magenta, yellow. For Usage Over Time or High Water Mark reports, two colors can be specified: the first for the usage line (default: blue) and the second for the available line (default: red). For barcharts, the specified colors are assigned to subsets of data in the order that each appears in the report. You can also specify colors in hexadecimal RGB format. See Customizing Appearance of Graphical Report Output |
| -datastyle <n[,n]> | Set the style of lines of data. Select from -1 (solid), 0 (dashed), 1 (dotted), or 2 (dotted-dashed). Default: -1. |
| -delim <delimiter> | Specify a RIF-format field delimiter. <delimiter> can be an arbitrary string. The default delimiter is a tab. Any fields that contain the delimiter are quoted with double quotes. Do not use double quotes ("), single quotes ('), or a backslash (\) as a RIF delimiter. |
| -elim_false_denials <interval> | Request the elimination of “false” denials occurring within an interval of given number of seconds. If not explicitly specified, interval is 10 seconds. |
| -encoding <charset> | Use the specified character set encoding to produce textual output files. The machine on which the output file is viewed must have a font that can display the specified character set. |

Table 8-10 • FlexNet Manager for Engineering Applications Reporting Command-Line Parameters

| Parameter | Description |
|--|---|
| <code>-event_type <type></code> | Specify for each event type to include in report. Available event types are used, denied, queued, exception, and checkout. If event_type is not specified, all event types except checkout are reported. |
| <code>-enddate <date></code> | Set the ending date for report output. |
| <code>-feat_uniq <ID></code> | Distinguish features using one of these IDs. Available values for ID are name, name_ver, name_key, name_asset. |
| <code>-field "field_name [,col_width] [,col_justification]"</code> | Include a field with these specifications in a Raw report. Field names are listed in later table. For example, <code>-field "even, 10, left"</code> |
| <code>-filter <category> <operator> <value></code> | Specify filtering expression. Filter categories are listed in a later table. |
| <code>-filterperiod "start" "end"</code> | Specify a time period during the week to include in report output. The start and end time formats are d.hh:mm:ss, where d is 1-7 (Day 1 = Sunday, Day 7 = Saturday). Designating days and seconds is optional. Surround start and end with double quotes. |
| <code>-fontbold</code> | Use bold font for the labels, title, and legend of the graph. Must also set <code>-fontface</code> and <code>-fontsize</code> . |
| <code>-fontface </code> | Set the font face for the labels, title, and legend of the graph. Select from the fonts available to the JVM on the machine. Must also set <code>-fontsize</code> . |
| <code>-fontitalic</code> | Use italic font for the labels, title, and legend of the graph. Must also set <code>-fontface</code> and <code>-fontsize</code> . |
| <code>-fontsize <points></code> | Set the font size in points for the labels, title, and legend of the graph. Must also set <code>-fontface</code> . |
| <code>-gap_threshold <seconds></code> | Specify the minimum duration of a gap in license service that is reported in a Server Coverage report. |
| <code>-graphical_template <template></code> | Specify a template to use for the graphical configuration, with a value between 1 and 6. |
| <code>-host <host></code> | Specify the host name. Required. |
| <code>-html</code> | Request report output in HTML format. |
| <code>-input input_filename1 [input_filename2...]</code> | Specify the list of report log(s) to read. The <code>-input</code> parameter can be used once followed by several report logs. Required (or <code>-input_single</code>) |

Table 8-10 • FlexNet Manager for Engineering Applications Reporting Command-Line Parameters

| Parameter | Description |
|---|--|
| <code>-input_single <"file_name"></code> | Specify a single report log for input. Useful for specifying a file name containing white space on Windows. Surround file name with double quotes. |
| <code>-interval <period></code> | Specify Usage Over Time or High Water Mark summary period. Available values for summary period are none, minute, hour, day, week, month. Not valid for Usage Efficiency or Raw reports. |
| <code>-jpeg [<width> <height>]</code> | Request graphical report output in JPEG/HTML format. Optionally, set the dimensions of the JPEG images in pixels. Default: 600 x 600. |
| <code>-license_server</code> | Specify a license server name. This results in a report that uses report log data from the specified license server(s). If this parameter is not used, the report is run using report log data from all available license servers. To specify multiple license servers, use the parameter multiple times—for example: <code>-license_server ls1 -license_server ls2</code> |
| <code>-linewidth <width></code> | Set the thickness of lines in the graph in pixels. Affects both data and axes. Default: 1. |
| <code>-no_auto_reduce</code> | Used to circumvent automatic data reduction on Usage Over Time and High Water Mark graphs. Not recommended. |
| <code>-orderby <category></code> | Sort categories are f (feature) and b (breakdown). More than one sort category can be specified at a time. For example, <code>-orderby fb</code> sorts first by feature, then by breakdown category. |
| <code>-output <output_filename_base></code> | Specify the output file name, without a file extension. Required for Text, HTML, RIF, JPEG/HTML, and PNG/HTML output. |
| <code>-output_dir <directory></code> | Specify the output directory. Required. |
| <code>-password <password></code> | Specify the password to use. Required. |
| <code>-periodic</code> | Calculate Summary Barchart or Usage Summary for time periods. Use in conjunction with <code>-interval</code> period. |
| <code>-png [<width> <height>]</code> | Request graphical report output in PNG/HTML format. Optionally, set the dimensions of the PNG images in pixels. Default: 600 x 600. |
| <code>-rif</code> | Request report output in Report Interchange Format (RIF). |
| <code>-showzero</code> | Include entries for features with no usage. This parameter is available for Usage Over Time, High Water Mark, and Usage Summary reports. |

Table 8-10 • FlexNet Manager for Engineering Applications Reporting Command-Line Parameters

| Parameter | Description |
|---------------------------------------|--|
| <code>-startdate <date></code> | Set the beginning date for report output. |
| <code>-text</code> | Request report output in plain text format. |
| <code>-textcolor <color></code> | Set the text color of a the axes, labels, title, and legend of a graphical report. Select from: white, black, red, green, blue, cyan, magenta, yellow. Default: black. Specify colors in hexadecimal RGB format. See Customizing Appearance of Graphical Report Output |
| <code>-timeval_seconds</code> | Show time value in RIF files as UNIX seconds since 00:00:00 January 1, 1970 GMT. This parameter is available for Raw, Usage Efficiency, and Usage Over Time reports. |
| <code>-title <title></code> | Set the title on a graphical report. |
| <code>-user <user></code> | Specify the user name. Required. |
| <code>-vendor_daemon</code> | Specify a vendor daemon name. This results in a report that uses report log data from the specified vendor daemon(s). If this parameter is not used, the report is run using report log data from all available vendor daemons. To specify multiple vendor daemons, use the parameter multiple times—for example: <code>-vendor_daemon vd1 -vendor_daemon vd2</code> |
| <code>-xml</code> | Request report output in XML document. |

Some parameters are specific to particular report types. See [Figure 8-11](#) for more details.



Note • No parameter is shown for Raw reports. If you generate a Raw report from the command line, you can choose which fields to display. See [Raw Report Fields](#) for information.

Table 8-11 • Report Type parameters

| Report Type Parameter | Description |
|------------------------|---|
| <code>-coverage</code> | Calculate Server Coverage data. Syntax: <code>report Server -coverage</code> |

| Report Type Parameter | Description |
|-----------------------|--|
| -hwmot [na] | Calculate High Water Mark data. Use the na option to omit displaying in the report the line that shows available licenses. Syntax: report Summary -hwmot |
| -pud [na] | Generates Peak Usage and Denial data, showing the number of peak usage licenses within a specified summary period, and the number or denials - if any - within the same period. Use the na option to omit displaying in the report the line that shows available licenses. Syntax: report Summary -pud |
| -uot [na] [pct] | Calculate Usage Over Time or High Water Mark data. <ul style="list-style-type: none"> • Use the na option to omit displaying in the report the line that shows available licenses. • Use the pct option to display usage in percentages of available licenses instead of count. Syntax: report Summary -uot |
| -usage_efficiency | Calculate Usage Efficiency data—the incremental usage of licenses. Syntax: report Summary -usage_efficiency |
| -usage_summary | Calculate Usage Summary data. Syntax: report Summary -usage_summary |
| -uu | Calculate Unique User Report data. Syntax: report Summary -uu |

Breakdown Categories

The following breakdown categories are available for FlexNet Manager for Engineering Applications Reporting from the command line.

Table 8-12 • Breakdown Categories Available from the Command Line

| Breakdown Category | Description |
|--------------------|--|
| exp_date | Expiration date on FEATURE/INCREMENT line (string with wildcards). |
| feat_hostid | HOSTID on FEATURE/INCREMENT line. |

Table 8-12 • Breakdown Categories Available from the Command Line

| Breakdown Category | Description |
|----------------------|--|
| group | GROUP name in options file for vendor. If you break down usage by group, you should do one of the following: <ul style="list-style-type: none">• Ensure that no user belongs to more than one GROUP.• Filter the report data to restrict the scope of the report such that the output contains no users that belong to more than one GROUP. |
| user | User name. |
| user_display | User display name (UNIX only). |
| user_flexver | Product's FlexNet licensing version. |
| user_hostname | User host name. |
| user_ipaddr | User IP address (nnn.nnn.nnn.nnn). |
| user_platform | Platform the application was compiled to run on. Requires FLEXlm v5+ (called FlexNet Publisher from version 10.0) in lmgrd, vendor daemon, and application. For example, sun4_u5 (Solaris), i86_n3 (Win95/NT). |
| user_product_version | Product version used. |
| user_project | User LM_PROJECT environment variable (or Windows registry). |
| user_vendor_def | Value of CHECKOUT_DATA passed by the application to the license server when the license was checked out. |
| vendor_string | VENDOR_STRING=text or "text" on FEATURE/INCREMENT line. |
| version | Version on FEATURE/INCREMENT line. |

Filter Categories

Table 8-13 shows the filter categories supported by FlexNet Manager for Engineering Applications, as long as your report log contains data in the category you select. In general, the later the FlexNet Publisher version of the vendor daemon that wrote the report log, the more of these filter categories are available in the report log.

Table 8-13 • Filter Categories Available from the Command-Line Filter

| Category | Description |
|------------|--|
| asset_info | asset_info=text on FEATURE/INCREMENT line. |
| dist_info | st_info=text on FEATURE/INCREMENT line. |

Table 8-13 • Filter Categories Available from the Command-Line Filter

| Category | Description |
|----------------------|--|
| exp_date | Expiration date on FEATURE/INCREMENT line (string with wildcards). |
| feat_hostid | HOSTID on FEATURE/INCREMENT line. |
| feature | FEATURE name. |
| group | GROUP name in end-user options file for vendor. |
| host_group | HOST_GROUP name in end-user options file for vendor. |
| include | USER or GROUP that is INCLUDED in the vendor's options file. |
| issued | ISSUED=dd-mm-yy on FEATURE/INCREMENT line. |
| issuer | ISSUER=text on FEATURE/INCREMENT line. |
| license_key | License key on FEATURE/INCREMENT line. |
| notice | NOTICE=text on FEATURE/INCREMENT line. |
| pkg_component | If 'Y', report only components from a SUITE PACKAGE. |
| pkg_suite | If 'Y', report only enabling feature from a SUITE PACKAGE. |
| reserve | USER or GROUP that is RESERVED in the vendor's options file. |
| serial | SN=nnn on FEATURE/INCREMENT line. |
| user | User name. |
| user_display | User display name (UNIX only). |
| user_flexver | Product's FlexNet licensing version. |
| user_hostname | User host name. |
| user_info | user_info=text on FEATURE/INCREMENT line. |
| user_ipaddr | User IP address (nnn.nnn.nnn.nnn). |
| user_platform | Platform the application was compiled to run on. Requires FLEXlm v5+ (called FlexNet Publisher from version 10.0) in lmgrd, vendor daemon, and application. For example, sun4_u5 (Solaris), i86_n3 (Win95/NT). |
| user_product_version | Product version used. |
| user_project | User LM_PROJECT environment variable (or Windows registry). |

Table 8-13 • Filter Categories Available from the Command-Line Filter

| Category | Description |
|-----------------|--|
| user_vendor_def | Value of CHECKOUT_DATA passed by the application to the license server when the license was checked out. |
| vendor_string | VENDOR_STRING=text or "text" on FEATURE/INCREMENT line. |
| version | Version on FEATURE/INCREMENT line. |

Raw Report Fields

[Table 8-14](#) shows the fields available in a Raw report from the command line or in a configuration file. These fields cannot be selected in the FlexNet Manager for Engineering Applications interface.

When running a raw report, you can specify all of the options at the command line, or you can create a configuration file and specify the location of the configuration file at the command line. See the section, [Using a Configuration File](#), for more information.

Table 8-14 • Raw Report Fields Available at the Command Line

| Field Name | Description |
|-----------------|---|
| -count | Number of licenses in request. |
| -duration | Amount of time event lasted. |
| -event | Event type. |
| -event_end_time | Event end time. |
| -event_time | Event start time. |
| -group | GROUP name in end-user options file for vendor. |
| -handle | An identifier assigned by a vendor daemon to a license checkout. This identifier is not unique because handles are recycled and reused by the vendor daemon. When correlating shared usage, it is helpful to show the handle and shared handle for each event so it is easy to spot which licenses were shared. |
| -host_group | HOST_GROUP name in end-user options file for vendor. |
| -shared_handle | An identifier assigned by a vendor daemon for a previous license checkout with which this checkout was shared. This identifier is not unique because handles are recycled and reused by the vendor daemon. When correlating shared usage, it is helpful to show the handle and shared handle for each event so it is easy to spot which licenses were shared. To display shared handle values, include checkout events in the report. |

Table 8-14 • Raw Report Fields Available at the Command Line

| Field Name | Description |
|-----------------------|--|
| -shared_group | An identifier assigned by a vendor daemon representing a group of one or more license checkouts which were satisfied with the same license. This identifier is not unique because the shared group identifiers are recycled and reused by the vendor daemon. |
| -user | User name. |
| -user_display | User display name (UNIX only). |
| -user_flexver | Product's FlexNet Licensing version. |
| -user_hostname | User host name. |
| -user_ipaddr | User IP address (nnn.nnn.nnn.nnn). |
| -user_platform | Platform on which the application was compiled to run. Requires FLEXlm v5+ (called FlexNet Publisher from version 10.0) in lmgrd, vendor daemon, and application. |
| -user_product_version | Product version used. |
| -user_project | User LM_PROJECT environment variable (or Windows registry). |
| -user_vendor_def | Value of CHECKOUT_DATA passed by the application to the license server when the license was checked out. |
| -vendor_string | VENDOR_STRING= <i>text</i> or " <i>text</i> " on FEATURE/ INCREMENT line. |
| -why | Reason a license was denied or checked in. |

9

Index

A

- aggregation
 - denial folding [44](#)
- aggregation usage data maintenance [58](#)

B

- best practices
 - usage data maintenance [58](#)
- Breakdown Categories [97](#)
- bulk importing, report logs [26](#)

C

- Classic reporting
 - data model [65](#)
- cleanup service [58](#)
- Cognos documentation [8](#)
- colors, specifying [89](#)
- command line
 - importing organizational structure XML [55](#)
 - running reports from [84](#)
- command-line arguments
 - running reports [92](#)
- Configuration File, loading [91](#)
- configuring LUM license servers [62](#)
- copying report configurations [18](#)

D

- Data Aggregation [44](#)
- data aggregation [9](#)
- data cleanup service [58](#)

- log messages [59](#)
- denial folding [44](#)
- documentation
 - Cognos [8](#)

E

- enabling report logs [15](#)

F

- Field Filters [87](#)
- Filter Categories [98](#)
- Flexera Analytics
 - predefined reports [44](#)
- format specification
 - RIF [14](#)

G

- gap [40](#)
- generating reports [26](#)
- Graphical Report Output, customizing appearance of [89](#)
- graphs
 - see report, usage over time
- group, in raw text reports [41](#)

H

- high water mark report [29](#)

I

- IBM LUM

- reporting on [61](#)
- importing organizational structures XML file [55](#)
- importorgstructure utility [55](#)
- interface
 - reporting module [11](#)

L

- license file
 - options file path [16](#)
- license limit, usage above [35](#)
- license usage, overdraft [35](#)
- LM_PROJECT environment variable [55](#)
- log messages
 - data cleanup service [59](#)
- LUM
 - configuring license servers to log events [62](#)
 - reporting on [61](#)
- LUM report logs
 - processing [62](#)

M

- managing report logs [16](#)

O

- options file, enable a report log [15](#)
- organizational structures [47](#)
 - adding using XML [48](#)
 - defining using FlexNet Manager for Engineering interface [56](#)
 - defining using XML [48](#)
 - importing XML [55](#)
 - sample XML for hosts [54](#)
 - sample XML for projects [55](#)
 - sample XML for users [54](#)
 - updating using XML [50](#)
- output formats [13](#)
- output, report interchange format [15](#)
- overdraft license usage [35](#)

P

- parameters
 - processLUMFile [63](#)
- peak usage and denial report [30](#)
- post-processing, reports [18](#)
- processing LUM report logs [62](#)
- processLUMFile
 - parameters [63](#)
- processLUMFile [62](#)
- project-based usage
 - enabling reporting [55](#)

R

- raw report [41](#)
- Raw Report Fields [100](#)
- Raw Report, defining [91](#)
- redundant servers, report logs [16](#)
- report
 - generating [26](#)
 - high water mark [29](#)
 - overdraft license usage [35](#)
 - raw [41](#)
 - raw text [41](#)
 - server coverage [39](#)
 - summary barchart [31](#)
 - types of [26](#)
 - usage above license limit [35](#)
 - usage efficiency [34](#)
 - usage over time [28](#)
 - usage summary [35](#)
 - viewing [26](#)
- report configurations
 - creating [12](#)
- Report Configurations screen [11](#)
- Report Database Cleanup [15](#)
- report interchange format [15](#)
- report log
 - bulk import [26](#)
 - enabling [15](#)
 - from redundant servers [16](#)
 - managing [16](#)
- report log rotation
 - scheduling [16](#)
- report log timestamps [18](#)
- report log usage data maintenance [57](#)
- report output formats [13](#)
- Report, level of detail [86](#)
- reporting
 - on IBM LUM [61](#)
 - schema [66](#)
- reporting module user interface [11](#)
- reporting workflow [8](#)
- reports
 - running from command line [84](#)
- reports and time zones [18](#)
- Reports tab [11](#)
- reports, specifying post-processing options [18](#)
- RIF format specification [14](#)
- RPT_CATALOG_SECTION Table [67](#)
- RPT_ERROR Table [71](#)
- RPT_FEATURE Table [67](#)
- RPT_REASON Table [70](#), [73](#)
- RPT_USAGE_LIC_DEVENT_153 Table [79](#)
- RPT_USAGE_LIC_EVENT_153 Table [67](#)
- RPT_USAGE_NOLIC_EVENT Table [71](#)

RPT_USAGE_NOLIC_EVENT_153 Table 77
RPT_VENDOR_DAEMON Table 70

updating organizational structures 50

S

sample report data 17
sample XML
 host organizational structure 54
 project organizational structure 55
 user organizational structure 54
sample.rl 17
scheduling
 report log rotation 16
server coverage report 39
special characters 51
summary barchart report 31

T

time zones 18
timestamps
 in report logs and reports 18
types of reports 26

U

usage data maintenance 57
 aggregation data 58
 best practices 58
 report log 57
usage data, bulk import 26
usage efficiency report 34
 see report, usage efficiency
usage over time report 28
 see report, usage over time
usage summary report 35
user interface
 reporting module 11

V

viewing reports 26

W

workflow 8

X

XML
 adding organizational structures 48
 defining organizational structures 48
 special characters 51

