



SAMreport **Users Guide**

VERSION 3.6

JUNE 2005

COPYRIGHT NOTICE

Copyright © 1999-2005 Macrovision Corporation and/or Macrovision Europe Ltd. All Rights Reserved.

The information contained herein contains confidential information and proprietary trade secrets and is the property of Macrovision Corporation. Any use, copy, publication, distribution, display, modification, or transmission of the information in whole or in part in any form or by any means without the prior express written permission of Macrovision Corporation is strictly prohibited. Except when expressly provided by Macrovision Corporation in writing, possession of this information shall not be construed to confer any license or rights under any of Macrovision Corporation's intellectual property rights, whether by estoppel, implication, or otherwise.

ALL COPIES OF THE INFORMATION, IF ALLOWED BY MACROVISION CORPORATION, MUST DISPLAY THIS NOTICE OF COPYRIGHT AND OWNERSHIP IN FULL.

TRADEMARKS

Macrovision[®], Globetrotter[®], SafeCast[®], FLEX^{lm}[®], FLEXnet[™], FLEXnet Publisher[™], FLEXnet Manager[™], FLEXnet Connector[™], FLEXenabled[™], and FLEXnet Certified[™] are registered trademarks or trademarks of Macrovision Corporation in the United States of America and/or other countries.

All other brand and product names mentioned herein are the trademarks and registered trademarks of their respective owners.

RESTRICTED RIGHTS LEGEND

The software and documentation are "commercial items," as that term is defined at 48 C.F.R. §2.101, consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.2702, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §227.2702-1 through 227.2702-4, as applicable, the commercial computer software and commercial computer software documentation are being licensed to U.S. government end users (A) only as commercial items and (B) with only those rights as are granted to all other end users pursuant to the terms and conditions set forth in the Macrovision Corporation standard commercial agreement for this software. Unpublished rights reserved under the copyright laws of the United States of America.

Printed in the USA.

May 2005

Contents

Preface	vii
Chapter 1 Introduction to SAMreport	11
1.1 Starting SAMreport	13
1.2 Stopping SAMreport	14
1.3 Tour of the SAMreport Interface	15
Chapter 2 Basics of Generating a Report	21
Chapter 3 Types of Reports	25
3.1 Usage Over Time Report	25
3.2 High Water Mark Report	27
3.3 Summary Barchart Report	29
3.4 Usage Efficiency Report	31
3.5 Usage Summary Report	33
3.6 Server Coverage Report	38
3.7 Raw Report	40
Chapter 4 Report Input and Output	47
4.1 Specifying Input Report Logs	47
4.2 Displaying Output to the Viewer Window	48
4.3 Navigating the SAMreport Viewer Window	48
4.4 Displaying Overdraft License Availability and Usage	50
4.5 Selecting Output Type and Output File Names	51
4.6 Output in Text Format	52
4.7 Output in HTML Format	52
4.8 Output in JPEG/HTML or PNG/HTMLFormat	52
4.9 Output in Report Interchange Format	53
Chapter 5 Output Arrangement and Detail	55
5.1 Summary Period	55
5.2 Breakdown Category	56
5.3 Sort Order Specification	57
5.4 Feature Description	58

5.5	Duplicate Grouping	59
5.6	Reporting on Usage by Project	60
Chapter 6	Data Filters	61
6.1	Defining an Event Filter	61
6.2	Defining a Time Period Filter	64
6.3	Reports and Time Zones	66
6.4	Selecting an Event Type Filter	66
6.5	Eliminating “False” Denials	68
6.6	Showing Zero Usage	70
6.7	Setting Start and End Dates for a Report	70
Chapter 7	Configuration Files	71
Chapter 8	Generating Command-Line Reports	73
8.1	Starting with the <i>SAMreport</i> GUI	73
8.2	Starting with the Command Line	74
8.3	Generating a Report from the Command-Line	75
8.4	Specifying a Different License File	76
8.5	Defining Input and Output Files	76
8.6	Defining the Level of Detail in a Report	77
8.7	Setting Breakdown Masks	78
8.8	Aliasing Feature or User Names in Report Output	79
8.9	Defining Field Filters	81
8.10	Customizing Appearance of Graphical Report Output	83
8.11	Loading a Configuration File	86
8.12	Defining a Raw Report	87
Appendix A	Enabling and Managing Report Logs	89
A.1	Finding a Report Log	89
A.2	Identifying a Report Log	90
A.3	Turning on a Report Log	91
A.4	The Size of the Report Log	94
A.5	Flushing Data to a Report Log	95
A.6	ssreport: A Report Log Utility	95
A.7	Managing Report Logs	96
A.8	Avoiding Corruption in Report Logs	101
Appendix B	Sample Scripts	103
B.1	last_month.csh	103
B.2	rotate_log.csh	106

Appendix C	Launcher and Java Options	109
C.1	Options for the <i>SAMreport</i> Launcher on Windows	109
C.2	Specifying Options to the Java VM	110
C.3	Specifying Initial and Maximum Java Heap Size	110
C.4	Switching Java Versions	111
Appendix D	Command-Line Report Reference	113
D.1	Command-Line Arguments	113
D.2	Breakdown Categories	119
D.3	Filter Categories	121
D.4	Raw Report Fields	123
Appendix E	<i>SAMreport</i> Error Messages	127
Index	131

Preface

This preface contains introductory information about license management, a description of *SAMreport*, and explanatory information about this guide. This guide assumes that you have already installed *SAMreport* with a production license.

Intended Audience

SAMreport and this manual are intended for use by system administrators, system analysts, and managers to produce usage reports for your software with FLEXnet Licensing or *SAMwrap* license management.

About This Manual

This manual is divided into eight chapters and five appendixes:

- Chapter 1, “Introduction to *SAMreport*,” introduces *SAMreport*.
- Chapter 2, “Basics of Generating a Report,” contains the basics of generating reports from the *SAMreport* interface.
- Chapter 3, “Types of Reports,” describes the types of reports available in *SAMreport*.
- Chapter 4, “Report Input and Output,” shows how to specify input and output files and describes output formats.
- Chapter 5, “Output Arrangement and Detail,” describes how to specify the level of detail in a report and how to arrange report output.
- Chapter 6, “Data Filters,” shows how to filter report data.
- Chapter 7, “Configuration Files,” describes configuration files.
- Chapter 8, “Generating Command-Line Reports,” gives instructions about generating reports from the command line.
- Appendix A, “Enabling and Managing Report Logs,” provides guidelines for enabling and managing report logs

What is License Management?

- Appendix B, “Sample Scripts,” provides sample scripts that may give you ideas about how to generate reports in `cron` jobs.
- Appendix C, “Launcher and Java Options,” describes Java options you can specify when launching *SAMreport*.
- Appendix D, “Command-Line Report Reference,” is a command-line reference for arguments to the `report` command.
- Appendix E, “SAMreport Error Messages,” lists some errors that you might get while running *SAMreport*.

This manual contains information on using *SAMreport* on Windows and UNIX. There are a few instances in this document where platform-specific instructions are listed. These instructions are identified with the platform name (Windows/UNIX) heading the information.

What is License Management?

License management is a set of technologies that monitors the use of software. In some cases it may also limit use of a product or record usage of a product.

Most software companies have legal agreements which detail the obligations of the manufacturer and user to each other. These agreements are often referred to as “licenses.” Some companies also include license management technology to simplify the legal use of their products. Throughout this documentation the term “license” refers to a “token” obtained through some electronic technology, not the legal agreement between a customer and a software vendor.

License management may be embedded into a software product by the software vendor, as in the case of FLEXnet Licensing, or “wrapped” around software that lacks license management, as in the case of *SAMwrap*. *SAMwrap* is license management software for system administrators to monitor and secure the use of software that does not offer embedded license management technology. Unless otherwise noted, all statements about FLEXnet Licensing are true for *SAMwrap* as well.

What is SAMreport?

SAMreport, a member of the Macrovision family of Software Asset Management (SAM) Solutions, is a report generator that helps you monitor the usage of applications that use FLEXnet Licensing or SAMwrap. *SAMreport* complements FLEXnet Licensing and SAMwrap by providing a graphical user interface (GUI) from which to generate a variety of license usage reports. *SAMreport* can help you make better decisions about your software assets.

Other Manuals

The *FLEXnet Licensing End User Guide* is on our web site at <http://www.macrovision.com/services/support/>.

To see a list of available online documents:

WINDOWS

Select Start | Programs | SAMreport | Documentation.

You can print a PDF version of the *SAMreport Users Guide* from:

```
C:\Program Files\Macrovision\samreport\version\machind\
  doc\UsersGuide.pdf
```

UNIX

Open `install_dir/Macrovision/samreport/current/htmlman/index.html`.

You can print a PDF version of the *SAMreport Users Guide* from:

```
install_dir/Macrovision/samreport/current/machind/doc/
  UsersGuide.pdf
```

Typographic Conventions

The following typographic conventions are used in this manual:

- The first time a new term is used it is presented in *italics*.
- Commands and path, file, and environment variable names are presented in a `fixed_font`.
- Other variable names are in an *italic_fixed_font*.
- Interface elements are in a sans-serif font.

Contacting Technical Support

Technical Support is available to customers and prospects with current support contracts. Please include the following information with your inquiry:

- Product Name
- Product Version
- Operating System Name and Version

Contact the support center for your area:

<http://www.macrovision.com/services/support/>

Introduction to *SAMreport*

SAMreport generates reports based on the license usage activity recorded in FLEXnet Licensing v2.4 to v10.8 report logs. FLEXnet Licensing report logs are produced by vendor daemons and maintain a record of the activity of the vendor daemon. *SAMreport* does not generate reports from debug log files, the standard output from `lmgrd`. Appendix A, “Enabling and Managing Report Logs,” describes how to start and manage report log files.

The instructions for using *SAMreport* are platform specific. On Windows, you run *SAMreport* from the Start menu or from the command prompt. On UNIX, you run the *SAMreport* report generator from the command line. If you are using *SAMsuite*, it is possible to start *SAMreport* from the *SAMconsole* window; we recommend using the command line. This document contains instructions for using *SAMreport* on Windows, on UNIX, and as part of *SAMsuite*, which is only available on UNIX. The screen shots in this document were taken using the Windows version of *SAMreport*. If you are using *SAMreport* on UNIX, the screen shots may be slightly different than what you see.

The *SAMreport* report generator comes with predefined report types which allow you to quickly start generating reports. The reports either summarize usage data (Summary reports), show individual usage events (Raw reports), or document the time that a license server is up (Server reports). Summary reports show report data graphically (line or bar graphs) or textually. Raw and Server reports give only textual output.

SAMreport processes the data from one or more report log files and writes the results to a file and/or displays results on the screen. Depending on the type of report, its textual output can be saved to a file in text, HTML, and/or Report Interchange Format (RIF). RIF can be used to export report data to another software program, such as a spreadsheet. Graphical report output can be saved to files in JPEG/HTML or PNG/HTML format. JPEG/HTML and PNG/HTML report output can be viewed in an HTML browser.

The following table summarizes the types of reports and output formats that are available in *SAMreport*.

Table 1-1: Summary of Report Types and Output Formats Available in SAMreport

Report Family	Report Type	Output	Output Formats
Summary	Usage Over Time	Graphical Textual	Display RIF JPEG/HTML PNG/HTML
	High Water Mark	Graphical Textual	Display RIF JPEG/HTML PNG/HTML
	Summary Barchart	Graphical	Display JPEG/HTML PNG/HTML
	Usage Efficiency	Graphical Textual	Display RIF JPEG/HTML PNG/HTML
	Usage Summary	Textual	Display HTML Text RIF
Server	Server Coverage	Textual	Display Text
Raw	Raw	Textual	Display Text RIF

For all report types, you can specify the particular features, times, users, hosts, IP addresses, displays, vendor-defined strings, and event types (used, denied, etc.) to include in the report.

You can arrange and display report data in a variety of ways. You can examine license usage over different time intervals, and you can break the report down by user, host, IP address, display, vendor-defined string, or a combination. For Raw reports you can specify the columns of data that are displayed, and you can group data on the fields of your choice.

SAMreport allows you to save customized report configurations. You can save combinations of report parameters in configuration files that you can load into the *SAMreport* main window to produce reports interactively, or you can load configuration files to produce reports with *SAMreport* from the command line. (See Chapter 8, “Generating Command-Line Reports.”)

Details on customization options for reports can be found later in this Users Guide.

1.1 Starting SAMreport

Throughout this document, the path to the directory where you installed the current version of the *SAMreport* software is referred to as *SAMreport_home*. On Windows, the default location for *SAMreport_home* is `C:\Program Files\Macrovision\samreport\version`. On UNIX, *SAMreport_home* is `install_dir/Macrovision/samreport/current`. Platform-specific instructions for starting *SAMreport* are below. Passing options to the *SAMreport* Java VM are described in Appendix C, “Launcher and Java Options.”

WINDOWS

Select Start | Programs | SAMreport | SAMreport or double-click `report.exe` in the *SAMreport_home* folder to start the *SAMreport* interface.

To start *SAMreport* at the command line on Windows, open a command window, change to the *SAMreport_home* directory, and type:

```
report
```

UNIX

To start the *SAMreport* interface from the UNIX command line, go to the *SAMreport_home* directory and type:

```
./report &
```

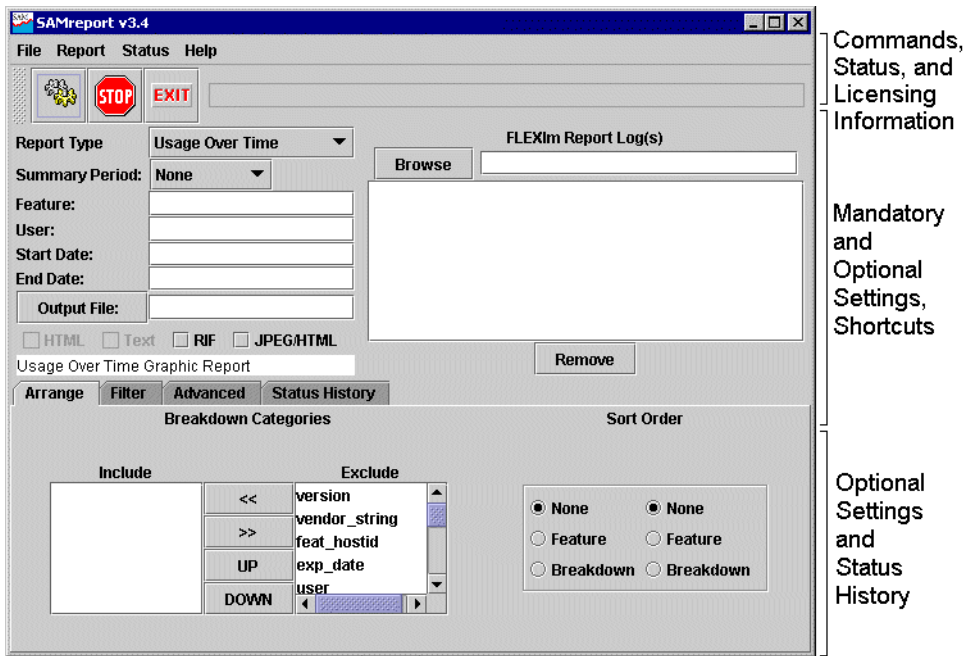
Stopping SAMreport

SAMSUITE

We recommend starting *SAMreport* from the command line as instructed above; although it is possible to start the *SAMreport* interface from the *SAMconsole* window in *SAMSuite*. Error diagnostics are available when you use the command line to start *SAMreport*.

To start the *SAMreport* interface in *SAMSuite*, select Reports | Reports from the *SAMconsole* window.

After a few seconds, the *SAMreport* logo screen appears; after a few more seconds, the *SAMreport* main window appears.



1.2 Stopping SAMreport

Select File | Exit or click the Exit button to stop the *SAMreport* interface.

1.3 Tour of the SAMreport Interface

The top portion of the *SAMreport* main window contains four menus:



- **File**
Commands to load or save report configuration files and to exit *SAMreport*. The command to exit is also found on a button on the toolbar.
- **Report**
Commands to start and stop generating reports. These commands are also found on buttons on the toolbar.
- **Status**
Commands to clear the entries in the Status History pane or to save the status entries to a file.
- **Help**
Help | About *SAMreport* shows the versions of *SAMreport* and Java that you're running. Note that the version of *SAMreport* is also displayed in the title bar.
Help | Show Licensing brings up a popup that displays vendor daemon and license server hostid for each license server that wrote each of the report logs in the FLEXIm Report Log(s) field.

The *SAMreport* toolbar contains three shortcut buttons, whose functions are also available in menus:



Generate Report



Stop Report Generator

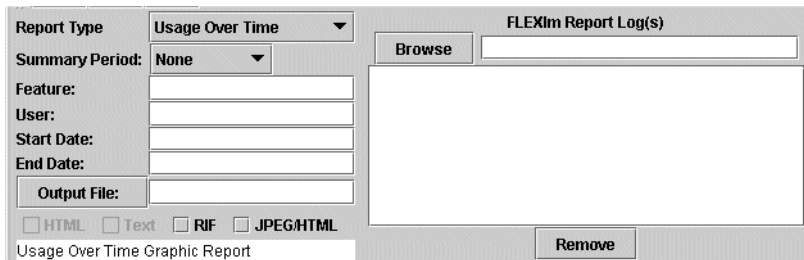


Exit *SAMreport*

Tour of the SAMreport Interface

To the right of the toolbar is a text box that displays the most recent status message. The history of status messages is available in the Status History pane.

Default settings that depend on report type are provided in the middle portion of the SAMreport main window.



The screenshot shows a software interface for configuring a report. On the left, there is a vertical list of fields: 'Report Type' (set to 'Usage Over Time'), 'Summary Period' (set to 'None'), 'Feature:', 'User:', 'Start Date:', 'End Date:', and 'Output File:'. Below these fields are four checkboxes: 'HTML', 'Text', 'RIF', and 'JPEG/HTML'. At the bottom left, the text 'Usage Over Time Graphic Report' is visible. On the right side, there is a section titled 'FLEXlm Report Log(s)' containing a 'Browse' button and a large empty text area. Below the text area is a 'Remove' button.

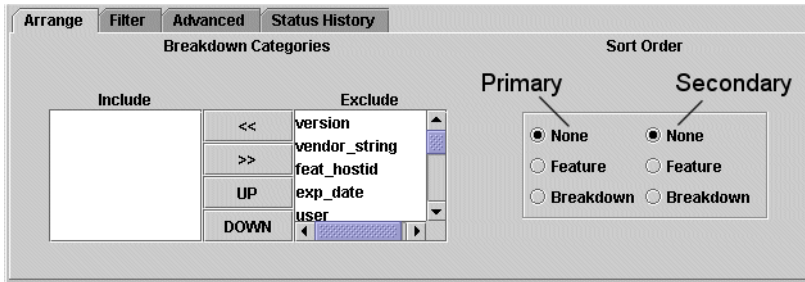
- The Browse field can be used to manually enter a path to a report log. The Browse button can be used to bring up a file selection browser to select a report log. The text area below the Browse field contains a list of the report logs that you have selected to generate a report.
- The Remove button removes a highlighted report log from the list of report logs that you have selected to generate a report.
- Report Type is a choice list from which you pick the type of report you would like to generate.
- Summary Period, Summarize By, or Summarize choice lists govern the way report data is summarized and displayed. Which (if any) of these choice lists is available depends on report type.
- Feature and User fields provide shortcuts for setting commonly used filters. Multiple feature or user entries can be separated by a space. Filters for feature, user, and other report event filters can be assigned in the Filter pane.
- Start Date and End Date fields can be used to limit the report to a subset of the time covered by a report log.
- The Output File field can be used to enter the path to the file name base of one or more output files. The Output File button can be used to bring up a file selection browser to select or assign a file name base for one or more output files. SAMreport adds a file name extension to the file name base, depending on the report format selected. An output file must be specified for any report that is generated in text, HTML, RIF, JPEG/HTML, or PNG/HTML format.

- Check boxes are used to select output file formats. Available formats are selectable; unavailable formats are not. PNG/HTML output cannot be specified in the GUI; it can only be specified in a configuration file or on the command line.

The bottom of the SAMreport main window contains four panes, shown like tabbed file folders. When shortcut fields are used to specify summary settings or filters, notice that the applicable pane settings update automatically to reflect the shortcut settings.

- **Arrange**

The Arrange pane is used to set breakdown categories and primary and secondary sort order specifications for a report. Some of these specifications can also be made in the Summarize By choice lists.

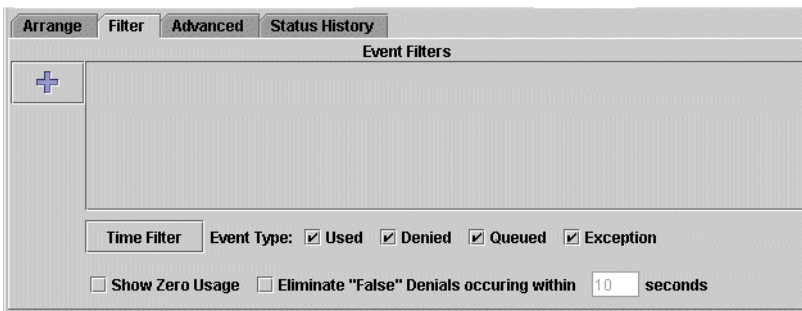


- **Filter**

The Filter pane is used to establish filters to limit the scope of a report.

- Event Filters — Restrict report data to events with desired characteristics
- Time Filter — Restrict report data to certain time periods each week
- Event Type Filter — Include exceptions, queued, denied, and/or used license requests in the report
- Show Zero Usage — Include features that have no usage recorded in the input report log in the report (available for Usage Over Time, High Water Mark, Summary Barchart, and Usage Summary reports)

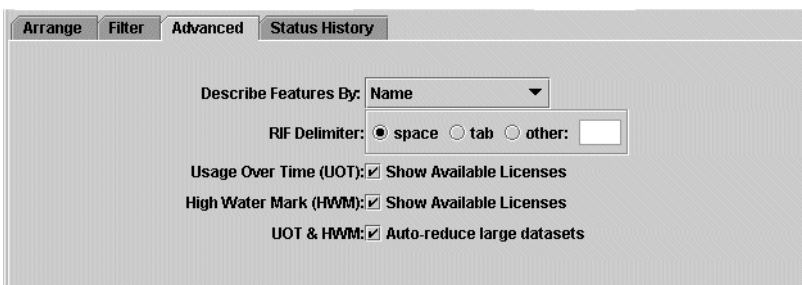
- Eliminate “False” Denials — Eliminate extraneous or “false” denial events that are logged by the license server as it processes checkout requests



- **Advanced**

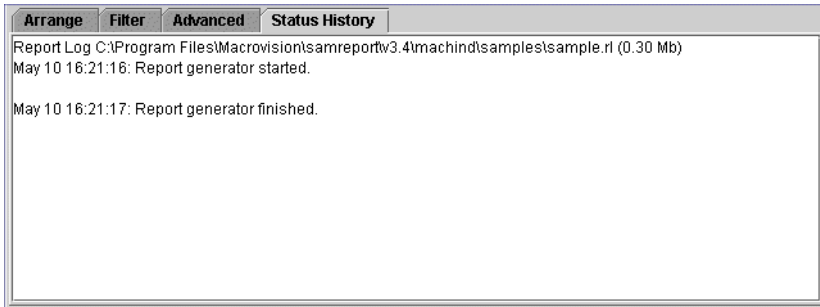
The Advanced pane contains a selection of infrequently used options:

- Describe Feature By — Use it to specify how features are described, whether by name or by a combination of name and another property.
- RIF Delimiter — Use it to configure field separators for RIF reports. Do not use double quotes (“), single quotes (’), or a backslash (\) as a RIF delimiter.
- Usage Over Time (UOT) and High Water Mark (HWM) — Use these to control scaling and data reduction applied to Usage Over Time and High Water Mark reports.



- **Status History**

The Status History pane contains status messages showing the progress of the report generator and any diagnostic messages.



Tour of the SAMreport Interface

Basics of Generating a Report

To generate a report from the *SAMreport* interface:

1. Start *SAMreport*.
2. Select a Report Type. *Mandatory*.

The type of report you select from the Report Type choice list governs what other selections are available. Descriptions of each report type are found in Chapter 3, “Types of Reports.”

3. Specify one or more input files. *Mandatory*.

You must specify the location of each of the report logs you want to run with *SAMreport* using one of the following two methods.

- Type the path to the report log in the text field to the right of the Browse button. Press Enter. Repeat this process for each report log that you want to add.
- Click the Browse button to select a report log from your system. Double-click the Open button in the File Selection box to add the selected report log file to your report. You can repeat this process to select more than one report log from the browser. If you are using Java 1.3+, multiple selection is supported.

After you select a report log, the report log file name appears in the text box above the Remove button. To remove a report log from the list, click to highlight the report log, then click the Remove button.

There is no fixed limit on the size of a report log that you run through *SAMreport*, but the execution time of a report is proportional to the amount of report log data that is processed. Improved processing speed can be achieved by configuring the JRE (see the *SAMreport Release Notes*), specifically when generating Summary reports from large report logs.

4. Specify an output format and/or an output file name base. *Mandatory for some report types.*

Different output formats are available, depending on the report type selected. Unavailable output formats are grayed out. All reports written to files require an output file name base and at least one output format selection.

The file name base can be entered in one of two methods:

- Type the path and file name base in the text field to the right of the Output File button. If no path is specified, the files are written to the current working directory.
- Click the Output File button to open a file selection browser. Find and double-click the directory in which you want to save the output file. Type the file name base in the file selection browser. Click the Save button.

Note: The appropriate file name extension is automatically appended to a file name base, depending on the files' output format.

5. Specify how you want report data summarized and arranged. *Optional.*
More information about summarizing and arranging report data can be found in Chapter 5, "Output Arrangement and Detail."
6. Specify how you want features described. *Optional.*
See Section 5.4, "Feature Description."
7. Select filters for report data. *Optional.*
See Chapter 6, "Data Filters."
8. Specify start and end dates for report output. *Optional.*
Use these fields to limit the date range of data to include in a report. To include data from the entire report log(s) in your report, leave these fields blank. See Section 6.7, "Setting Start and End Dates for a Report."

9. Select Report | Generate or click the Generate Report button.

A SAMreport Viewer window appears and the report progress indicator moves from 0–100% as the report is generated. Status messages, including any errors that occurred during report processing, are written in the Status History pane. You should see a status message indicating whether a report was produced. Even if a report was generated, there may be warnings or errors that are displayed here.

If you want to stop generating a report, click the Stop button on the toolbar or select Report | Stop.

After a report is generated, the SAMreport Viewer window contains one or more pages of the report output.

Types of Reports

SAMreport's report types belong to three report families: Summary reports, Raw reports, and Server reports. Summary reports summarize usage information about each feature in the report log(s). Raw reports show individual usage events in textual format. Server reports give information about a license server.

SUMMARY REPORTS

- Usage Over Time Report
- High Water Mark Report
- Summary Barchart Report
- Usage Efficiency Report
- Usage Summary Report

SERVER REPORTS

- Server Coverage Report

RAW REPORTS

- Raw Report

This section shows you how to generate reports using the *SAMreport* graphical user interface. See Chapter 8, “Generating Command-Line Reports,” for instructions for generating reports from the command line.

3.1 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 choice list. 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 Over Time Report

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. For each feature, two lines are plotted: one representing licenses available and one representing licenses used. The *Advanced* pane has a scaling option best used with features that have low levels of activity when compared with the number of licenses available. This option removes the *available* line from the generated graph. To use this option, uncheck the applicable Show Available Licenses box on the *Advanced* pane. If you generate a Usage Over Time or High Water Mark report from the command line (see Appendix D, “Command-Line Report Reference”), you can:

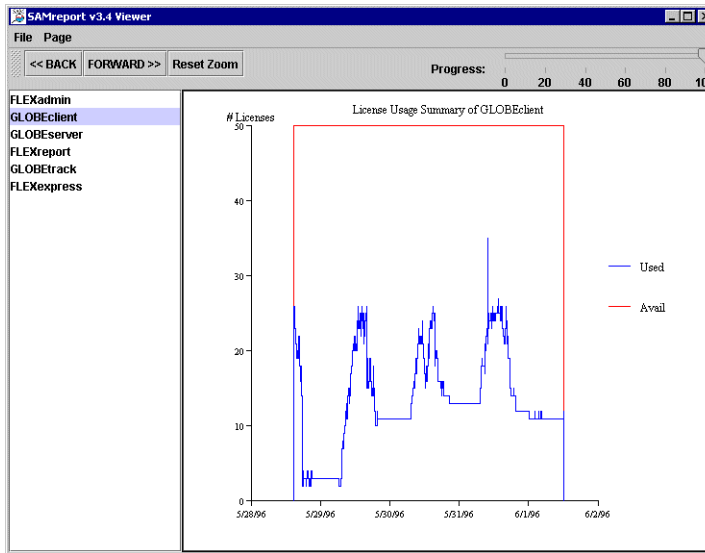
- Omit the line for available licenses by adding the optional argument `na`
- Display usage in percentages of available licenses (instead of number of licenses) by adding the optional argument `pct`

You can zoom in on data in Usage Over Time reports—see Section 4.3, “Navigating the SAMreport Viewer Window.”

To generate a Usage Over Time report:

1. Select Usage Over Time from the Report Type choice list.
2. Select a report log.
3. Select an output file type and output file name base, if applicable. Data for a Usage Over Time report can be written to a file in RIF format or files in JPEG/HTML or PNG/HTML format, as well as displayed as a line graph on the screen.
4. Select parameters for the report. You may select data filters and ways to summarize and order report data.

5. Click the Generate Report button. You see a SAMreport Viewer:



Note: The High Water Mark report can produce dense reports containing more points than can be displayed. Like the Usage Over Time report, these dense data sets are thinned automatically as needed.

3.2 High Water Mark Report

Like a Usage Over Time report, a 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.

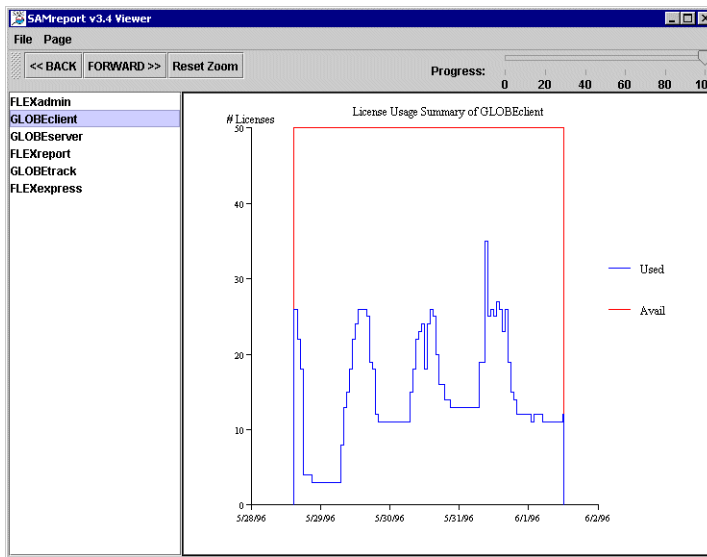
Each page of the report displays a line graph of a feature. The horizontal axis of the line graph is time, scaled to fit all the data for one feature into one window. The vertical axis is number of licenses used. For each feature, two lines are plotted: one representing licenses available and one representing licenses used. If you generate a Usage Over Time or High Water Mark report, you can omit the line for available licenses by unchecking the Show Available Licenses box on the Advanced pane or if you are using the command line,

High Water Mark Report

adding the `na` flag (see Appendix D, “Command-Line Report Reference”). You can zoom in on data in High Water Mark reports—see Section 4.3, “Navigating the SAMreport Viewer Window.”

To generate a High Water Mark report:

1. Select High Water Mark from the Report Type choice list.
2. Select a report log.
3. Select an output file type and output file name base, if applicable. Data for a High Water Mark report can be written to a file in RIF format and files in JPEG/HTML or PNG/HTML format, as well as displayed as a line graph on the screen.
4. Select parameters for the report. You may select data filters and ways to summarize and order report data.
5. Click the Generate Report button. You see a SAMreport Viewer:



Usage Over Time and High Water Mark Graphs are subject to automatic data reduction when datasets are particularly large. If automatic data reduction has occurred, you see a warning in the Status History window similar to:

```
"Warning: Graph for F1 from Jan 1, 1999 to Jan 1, 2001 is too large (99345 points).
```

```
Graph automatically reduced to 2014 points."
```

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

3.3 Summary Barchart Report

A Summary Barchart report provides a useful format for comparing license utilization across users, features, etc. 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 as 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)

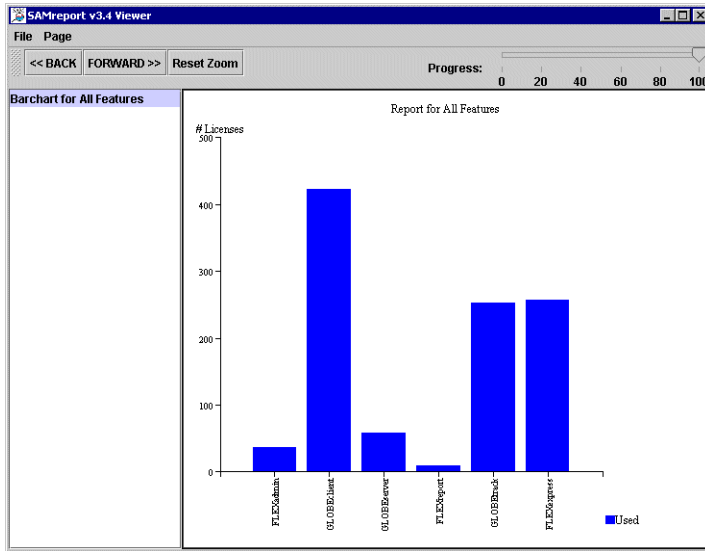
The bars are arranged in alphabetical order. The output of a Summary Barchart is displayed to the screen and can be saved to files in JPEG/HTML or PNG/HTML format.

To generate a Summary Barchart report:

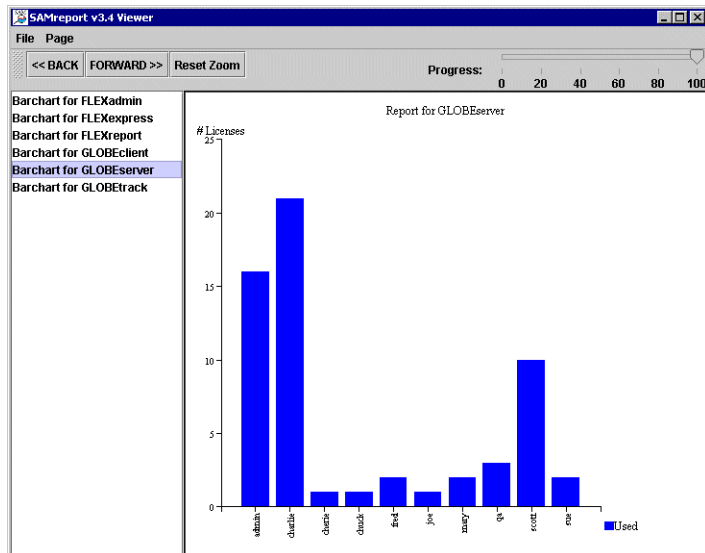
1. Select Summary Barchart from the Report Type choice list.
2. Select a report log.
3. Select parameters for the report. Select metric and summary settings from the Summarize and Summarize By choice lists. You may also select data filters. Features with no usage are not listed in the report unless you check the Show Zero Usage check box in the Filter pane.
4. Click the Generate Report button.

Summary Barchart Report

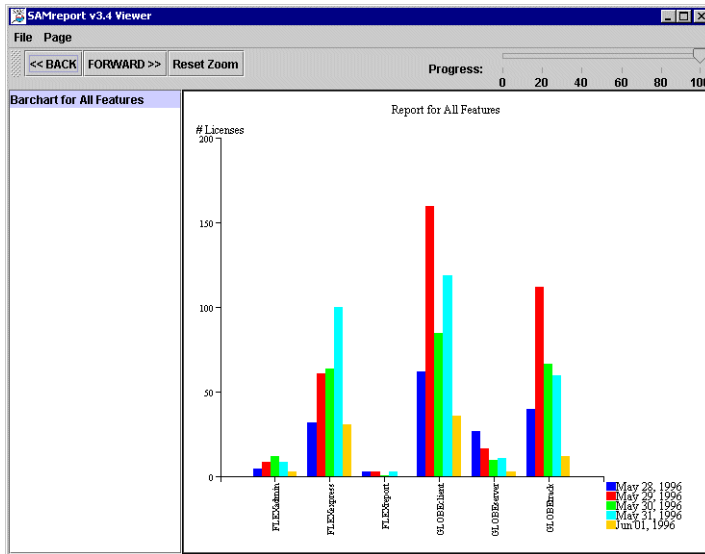
Here is an example of number of licenses used, summarized by Feature. This report compares the number of checkouts for each feature:



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



Here is an example of the same report with the number of licenses used summarized by Feature and summarized over each day in the report log. This report compares the daily usage levels for each feature over the reporting period:



3.4 Usage Efficiency Report

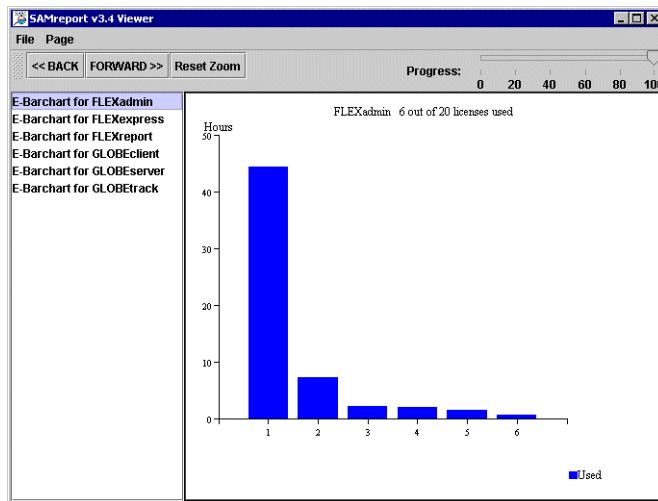
A Usage Efficiency report shows you how long each successive license for a given feature was in use. This kind of analysis may help you decide how many licenses you need for that feature. The report legend shows the maximum number of licenses checked out at one time compared to the total number of licenses available. The first bar in the graph shows how long the first license for that feature was checked out, the second 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.

Usage Efficiency Report

To generate a Usage Efficiency report:

1. Select Usage Efficiency from the Report Type choice list.
2. Select a report log.
3. Select an output file type. Data for a Usage Efficiency report can be written to a file in RIF format or files in JPEG/HTML or PNG/HTML format, as well as displayed as a barchart on the screen.
4. Select parameters for the report. There are no summary choice lists for a Usage Efficiency report.
5. Click the Generate Report button.

Here is an example of a Usage Efficiency report. Note that only six out of 20 available licenses were ever in use at one time; rarely were more than two licenses in use.



There are several special notations which 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(s)
- Vendor daemon crash that creates an unrecognizable `lm_log_start` in a pre-v5 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

3.5 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.

The header of a Usage Summary report provides a summary of information about the report log and the parameters of the report:

```
FLEXlm License Utilization Report date time
```

```
Produced by SAMreport version
```

```
Input Report Log:           List of input report log(s).
Output File Name:          Name of report output file.
Report Log(s) Start:       Earliest start date in any report log.
Report Log(s) End:         Latest end date in any report log.
Report Start Date:         Date to begin report—either a filter
                           start date or the starting date of the
                           earliest report log.
Report End Date:           Date to end the report—either a filter
                           end date or the ending date of the
                           latest report log.
```

Usage Summary Report

Uptime:	Duration in real clock time between the Report Start and End Dates that a license server was up.
Vendor:	Vendor daemon name(s).
License Server Host(s):	Host name(s) of the license server machine(s) in the report log(s) used to generate this report.
Sort by:	List of categories used to sort the report. May be none.
Breakdown by:	Report breakdown categories, if any.
Filter Start Date:	Date set to start report, if any. This field is omitted from the report if no start date is specified.
Filter End Date:	Date set to end report, if any. This field is omitted from the report if no end date is specified.
Event Filters:	List of event filters for the report. May be none.
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 this report.
False Denial Elimination:	Whether or not false denials were considered for elimination and if so, how many were eliminated.

A Usage Summary report with default settings contains one line per feature and 11 columns per line. From left to right in the report, the columns are:

```

=====
Time Period  --Available--  -----License Requests-----  ----Used----
              max    hours  total  used  que  den  unsup  %den  max  hours  %hrs
-----
Summary For: FLEXadmin
Total        20    1871    37    37    0    0    0    0    6    58.7  3.1
-----
Summary For: FLEXexpress
Total        50    4677    257   257    0    0    0    0    31   911.5 19.5
-----
Summary For: FLEXreport
Total        20    1871     9     9    0    0    0    0    2    23.6  1.3
=====

```

Table 3-1 describes the columns listed in a Usage Summary report:

Table 3-1: Columns in a Usage Summary Report

Category	Column Heading	Description
Available	max	Maximum number of available licenses. Includes overdraft licenses.
	hours	Number of available hours of license usage.

Table 3-1: Columns in a Usage Summary Report (cont.)

Category	Column Heading	Description
License Requests	total	Total number of license requests. (RIF format output does not include this column because it is just the sum of used + den + unsup.)
	used	Number of licenses requests fulfilled.
	que	Number of license requests queued.
	den	Number of license requests denied. If client application is built with FLEXlm v9.0+ (called FLEXnet Licensing from v10.0 on), ultimate denial events are not distinguished from other denial events in a Usage Summary report. For more detail on denials, see Section 3.7.2, "Displaying Denials in Raw Reports" and Section 6.5, "Eliminating "False" Denials."
	unsup	Number of license requests not fulfilled because the feature is unsupported. Also referred to as exception events.
	%den	Percent of license requests that were denials.
Used	max	Maximum number of licenses used simultaneously. Includes overdraft licenses.
	hours	Number of available license hours used.
	%hrs/%used	Percent of available license hours used.

To generate a Usage Summary report:

1. Select Usage Summary from the Report Type choice list.
2. Select a report log.
3. Select an output file type. Data for a Usage Summary report can be written to a file in HTML, text, or RIF format. A Usage Summary report is displayed to the screen from a file. Therefore, if you want to display a Usage Summary report to the screen, you must specify an output file format and output file name base.
4. Select parameters for the report. Select summary settings from the Summarize By choice lists. You may also select data filters. Features with no usage are not listed in the report unless you check the Show Zero Usage check box in the Filter pane.
5. Click the Generate Report button.

Here is an example of a Usage Summary report displayed to the screen from a text output file:

Text Summary: C:\MVS\exam

Features With Usage:
FLEXadmin, FLEXexpress, FLEXreport, GLOBEclient, GLOBEserver, GLOBEtrack

Features Without Usage: none

False Denial Elimination: OFF

```

=====
Time Period  --Available--  -----License Requests-----  ----Used----
              max      hours  total  used  que  den  unsup  %den  max  hours  %hrs
-----
Summary For: FLEXadmin
Total          20      1871      37   37   0   0   0   0   6   58.7  3.1
-----
Summary For: GLOBEclient
Total          50      4677     423  423   0   0   0   0   35  1269  27.1
-----
Summary For: GLOBEserver
Total           4       374      61   59   0   2   0   3   4  214.9  57.4
-----
Summary For: FLEXreport
Total          20      1871       9    9   0   0   0   0   2   23.6  1.3
-----
Summary For: GLOBEtrack
Total          50      4677     253  253   0   0   0   0   33  1207  25.8
-----
Summary For: FLEXexpress
Total          50      4677     257  257   0   0   0   0   31  911.5  19.5
-----

```

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

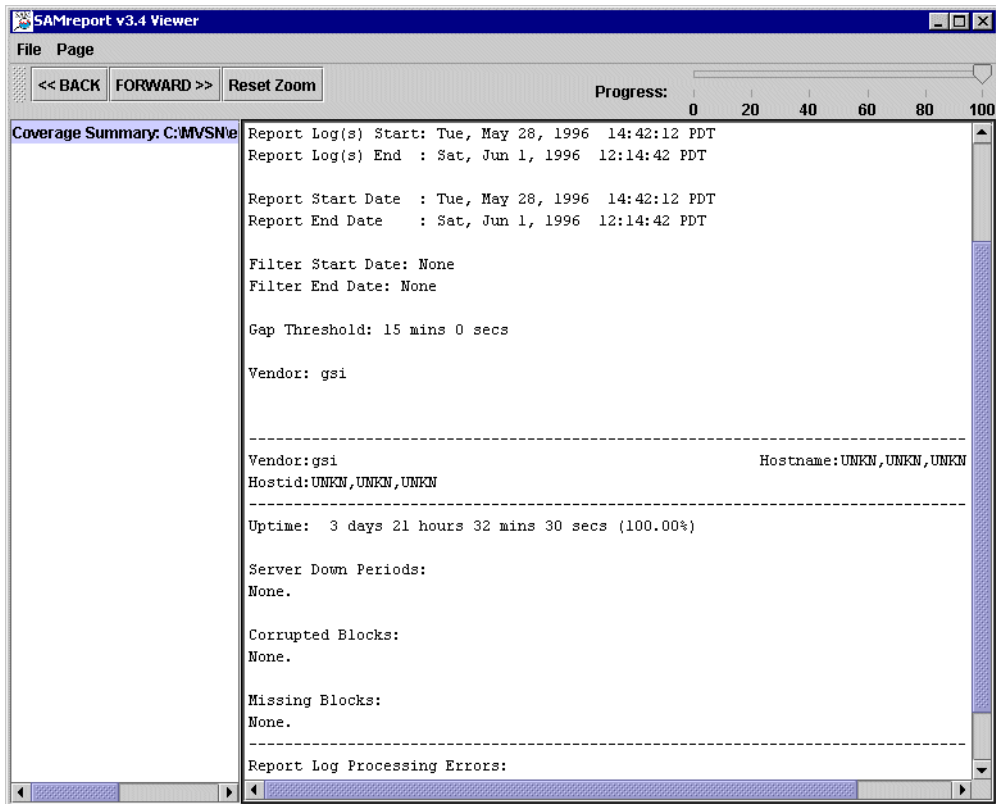
3.6 Server Coverage Report

The Server Coverage report documents the time during which the license server was serving 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. A different minimum gap in service can be used by generating the report at the command line and specifying a different `-gap_threshold` value in seconds.

To generate a Server Coverage report:

1. Select Server Coverage from the Report Type choice list.
2. Select a report log.
3. Specify an output file name base.
4. The Text output format is selected by default for you when you select the Server Coverage report; confirm it is selected.
5. Select parameters for the report. You can set start and end dates for the report output, but other settings for license usage reports are not applicable to Server Coverage reports.
6. Click the Generate Report button.

Here is an example of a Server Coverage report displayed to the screen in the SAMreport Viewer:



Note that 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. See Appendix E, “SAMreport Error Messages,” for details of possible errors.

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.

3.7 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 are intended to help troubleshoot license server activity. For example, you may want to analyze the nature of certain denial or exception events.

To generate a Raw report:

1. Select Raw from the Report Type choice list.
2. Select a report log.
3. Select an output file type. Data for a Raw report can be written to a file in text or RIF format. A Raw report is displayed to the screen from a file. Therefore, if you want to display a Raw report to the screen, you must specify an output file format and output file name base.
4. Select parameters for the report. There is no summary choice list for a Raw report.
5. Click the Generate Report button.

Here is an example of a Raw report displayed to the screen in the SAMreport Viewer:

Raw Report: C:\MVSN\examples

Event Filters: None
False Denial Elimination: OFF

```

=====
Event  User      Host      Display      #  Time              Duration      Group
-----
Events For: GLOBEclient
Used   sam        brick     ncd10:0.0    1  May 28, 1996 14:45:12 00d 00:16:03 UNKN
-----
Events For: GLOBEtrack
Used   sam        brick     ncd10:0.0    1  May 28, 1996 14:45:12 00d 00:16:03 UNKN
-----
Events For: FLEXexpress
Used   sam        brick     ncd10:0.0    1  May 28, 1996 14:45:12 00d 00:16:03 UNKN
-----
Events For: GLOBEclient
Used   qa         peppy     curly:0.0    1  May 28, 1996 14:44:16 00d 00:19:43 UNKN
-----
Events For: GLOBEtrack
Used   qa         peppy     curly:0.0    1  May 28, 1996 14:44:16 00d 00:19:43 UNKN
-----
Events For: FLEXexpress
Used   qa         peppy     curly:0.0    1  May 28, 1996 14:44:16 00d 00:19:43 UNKN
-----
Events For: GLOBEclient
Used   fred       bee       gt:0.0       1  May 28, 1996 14:45:39 00d 00:21:25 UNKN
=====

```

The columns in the above 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. This set of default fields is loaded into a Raw report generated through the GUI from the `SAMreport_home\machind\samples\raw.config` file.

Note: Raw reports are fully customizable when invoked from the command line (see Section 8.12, “Defining a Raw Report”). Many fields from the report log are available from the Raw report. See Section D.4, “Raw Report Fields,” for a list of available fields.

3.7.1 Displaying Duplicate Grouping Information in Raw Reports

Several fields can be added to Raw report output to help correlate checkouts related by duplicate grouping: `handle`, `shared_handle`, and `shared_group`.

In a Raw report, a Used event represents a checkout/checkin pair. It displays the checkout time (`event_time`), checkin time (`event_end_time`), and usage duration (`duration`). A Checkout event is the checkout from a license checkout/checkin pair. In a Raw report, a Checkout event can display the checkout time (`event_time`), as well as the shared handle on the checkout (`shared_handle`). The Checkout event type is not available in the *SAMreport* interface and is not included in any reports by default. Unless you are examining duplicate grouping, Checkout events provide information already provided by Used events, and by default, are not included in a report.

To display data in the `shared_handle` field, an additional event type, Checkout, must be specified at the command line or in the configuration file(s) used to run the Raw report. Remember that if any event type is specified explicitly, all the event types that you want to display in a report must be listed explicitly and individually.

```
...
-event_type used
-event_type queued
-event_type denied
-event_type exception
-event_type checkout
...
-field handle,10
-field shared_handle,10
-field shared_group,10
```

All license checkouts are assigned a handle. If a checkout shares a license with a previous checkout, as a result of duplicate grouping, the later checkout is assigned a shared handle that matches the previous checkout's handle. A shared handle is reported only on a checkout and not on a checkin because because the event to which the shared handle refers is guaranteed to exist only at the time of the checkout.

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

handle, shared_handle, and shared_group are identifiers that are re-used by a vendor daemon, and are not, therefore, unique.

Consider an application license that has duplicate grouping based on host.

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

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

Time 1	User A checks out license on Host A	A: handle 1 A: shared group 100
Time 2	User B checks out license on Host A	B: handle 2 B: shared handle 1 B: shared group 100
Time 3	User A checks in license on Host A	
Time 4	User C checks out license on Host A	C: handle 3 C: shared handle 2 C: shared group 100

By examining the Raw report output, you can see which checkouts are related through duplicate grouping by matching shared handle and handle values. The checkout times displayed in the Checkout events can be matched with the checkout times in the Used events; the durations in those Used events can be summed.

3.7.2 Displaying Denials in Raw Reports

After you install your production *SAMreport* license, you might want to generate a report with a sample configuration file included in the `machind\samples` directory called `rawdennied.config`. Run with one of your report logs, this configuration file gives you a Raw report of denials and the reason (`why` field) for each denial.

A denial indicates that a candidate license was not able to satisfy the checkout request. Possible explanations for denials in Raw reports are shown below.

All licenses are reserved for others.

The available licenses are being withheld because some of the licenses are reserved for other users in the options file.

Checkout exceeds MAX specified in options file.

The available licenses are being withheld because there is a MAX available for this feature set in the options file.

Checkout request rejected by vendor-defined checkout filter.

The application has a built-in test for the license that failed and rejected the license checkout.

Clock difference too large between client and server.

The license checkout request was denied because the time difference between the client and server was more than 24 hours.

Duplicate selection mismatch for this feature.

A different version of the application is in use and has specified Duplicate Grouping criteria that are different than those specified in the checkout request.

Feature has expired.

The license for this feature has expired; a new license will be needed.

Feature removed during lmreread or wrong SERVER line hostid.

Checkout failure due to two possible causes:

- The feature is removed during lmreread, but the FLEXenabled application is reading an old copy of the license file which still has the removed feature.
- The hostid on the SERVER line is for a different host, so all features in this license file were removed.

Feature start date is in the future.

The feature is not enabled yet. The current date is before the feature start date.

FLEXlm version of vendor daemon is too old.

The vendor daemon is built with an older version of FLEXnet Licensing than the client application. The daemon must be the same version or higher than FLEXnet Licensing in the application. A new vendor daemon can be obtained from the software vendor.

License BORROW support not enabled.

The license or the application does not support license borrowing, but borrowing was requested by a user.

License file does not support this version.

The version of the application is not supported by the license. A new license should be obtained from the software vendor.

Licensed number of users already reached.

All available licenses have been checked out.

License server does not support this version of this feature.

The version of the application is not supported. A new license should be obtained from the software vendor.

License server system does not support this feature.

The feature has expired on the server, has not yet started, or the version is greater than the highest supported version.

No licenses available to borrow.

There is no license available for the borrower to checkout. This only applies to v8+ of FLEXnet Licensing.

No such feature exists.

The feature could not be found in the license file.

Raw Report

Platform not enabled.

The software is running on a platform for which it is not authorized by Macrovision. Contact your vendor for further information.

Request for more licenses than this feature supports.

A checkout request was made for more licenses than are available.

Cannot borrow that long.

The user has specified a borrow period longer than the license allows.

This platform not authorized by license.

The license includes a PLATFORM keyword and this platform is not listed.

User/host on EXCLUDE list for feature.

The user/host is restricted from accessing a license because it is listed in the EXCLUDE list in the options file.

User/host not on INCLUDE list for feature.

The user/host is restricted from accessing a license because it is not listed in the INCLUDE list in the options file.

USER_BASED license has not specified users -- see server log.

The user-based licenses must have users specified in the options file.

Users are queued for this feature.

All available licenses have been checked out and there are other users in the queue for this feature.

If a client application has been built with FLEXlm v9.0+ (called FLEXnet Licensing from v10.0 on), Raw report output displays ultimate denial events separately from other denial events. Ultimate denial events are not distinguished from other denials in a Summary report.

Report Input and Output

4.1 Specifying Input Report Logs

You can create reports using one or more report logs from one or more FLEXnet Licensing vendor daemons, as long as each report log is written by a single vendor daemon running on a single host (each redundant vendor daemon must write to its own report log).

The only valid input files are FLEXnet Licensing report logs, which start with `lm_log_start` or `# FLEXlm Report Log`. These report logs may have been produced by redundant license servers, a colon-separated list of license servers, or they may simply be archived report logs from one or more license servers. Report logs must be locally accessible (via NFS) to *SAMreport*.

Appendix A, “Enabling and Managing Report Logs,” describes how to start and manage report log files.

You must specify at least one report log before generating a report. You must specify the location of each of the report logs you want to run with *SAMreport* using one of the following two methods.

- Type the path to the report log in the text field to the right of the Browse button. Press Enter. Repeat this process for each report log that you want to add.
- Click the Browse button and use the Report Log file browser to select a report log from your system. Double-click the report log file. You can repeat this process to select more than one report log from the browser.

After you select a report log, the report log file name appears in the text box above the Remove button. To remove a report log from the list, click to highlight the report log, then click the Remove button.

There is no fixed limit on the size of a report log that you run through *SAMreport*, but the execution time of a report is proportional to the amount of report log data that is processed.

4.2 Displaying Output to the Viewer Window

All reports generated in the *SAMreport* interface are displayed to the screen in a *SAMreport Viewer* window, even if you specify one or more output files.

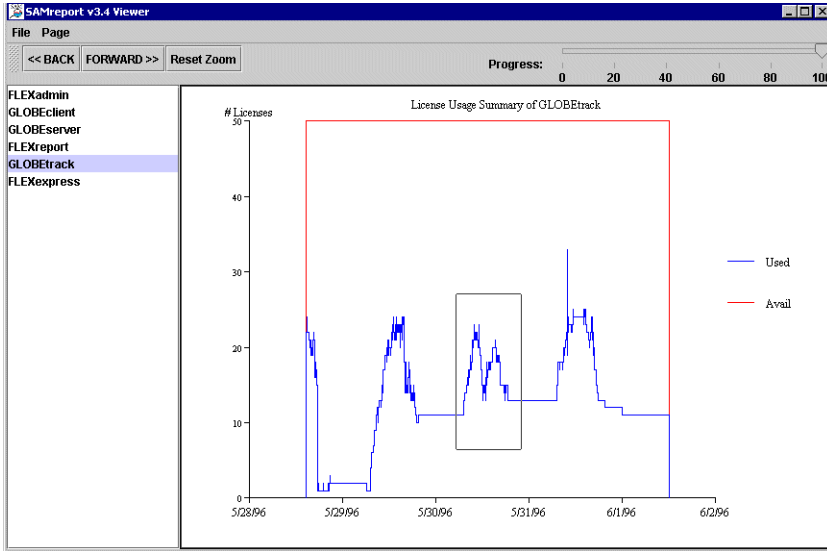
4.3 Navigating the *SAMreport Viewer* Window

The *SAMreport Viewer* window is divided into two sections. A list of the titles of individual pages of a report or a list of output formats is displayed in the left side of the *SAMreport Viewer*. A single graphical or textual page of a report is displayed in the right side of the *Viewer*. If the displayed page is longer than the display space (often the case with textual report output), you can read the rest of the page by moving the scroll bar at the right margin of the *Viewer*. A navigation toolbar and report progress indicator are displayed at the top of the *Viewer*.

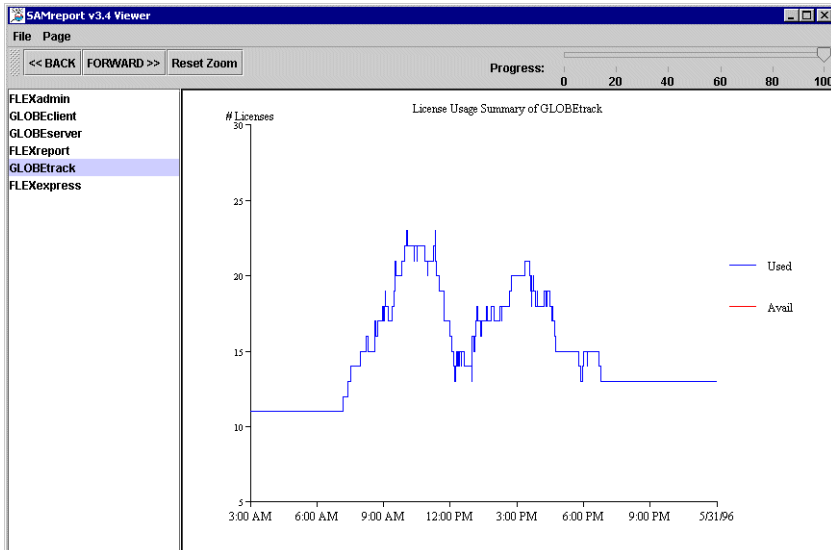
You can display different pages or output formats of a report in one of three ways:

- Click the page name in the left side of the window
- Click the <<BACK and FORWARD>> buttons in the toolbar
- Select Next or Previous from the Page menu

To zoom in on an area if a graphical Usage Over Time or High Water Mark report, sweep out a box with the left mouse button that contains the area of the graph that you want to enlarge.



When you release the mouse button, the Viewer displays the enlarged area of the graph.



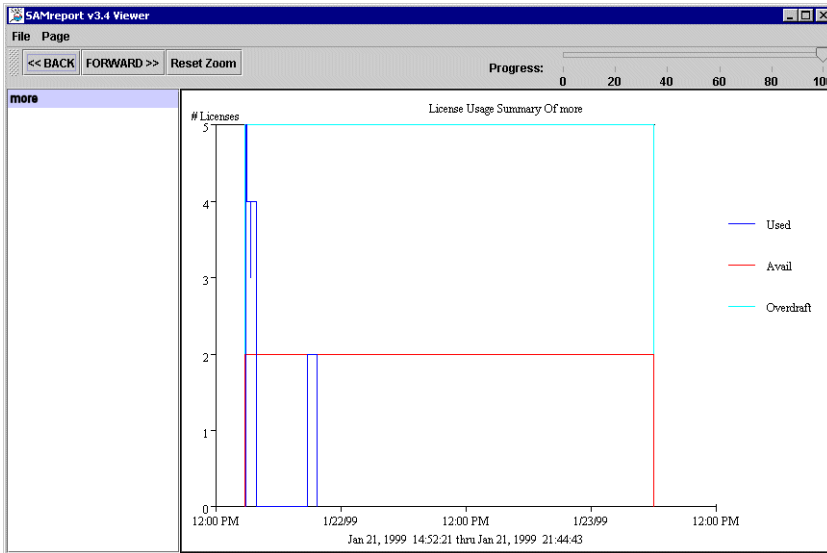
You can zoom in on an area several times.

You can save a zoomed image of a graphical report by selecting Page | Save As. Images are saved in JPEG format.

Click the Reset Zoom button to return to the original page display.

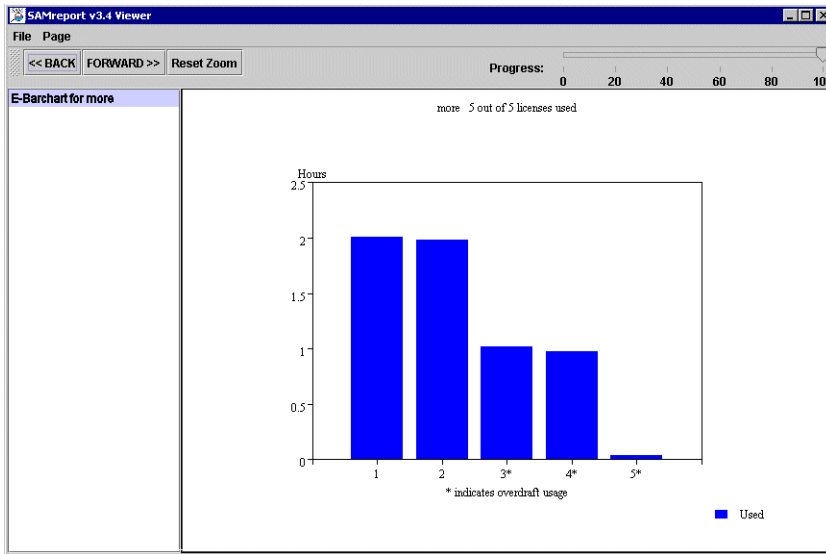
4.4 Displaying Overdraft License Availability and Usage

Your vendor may have issued you some overdraft licenses in addition to the regular licenses indicated by the count on the FEATURE line. The overdraft licenses for a feature are recognized by `OVERDRAFT=overdraft_count` in a FEATURE line. *SAMreport* displays the availability and/or usage of overdraft licenses differently depending on the type of report. You cannot differentiate overdraft from regular license usage with filters or summary breakdowns.



In a Usage Over Time or High Water Mark report, the number of regular available licenses, the total number of available licenses (with overdraft licenses added), and the number of licenses used are shown separately. The `na` option set at the command line for a Usage Over Time or a High Water Mark report suppresses both the line representing regular license availability and the line representing total license availability.

In a Usage Efficiency report, overdraft usage is denoted with an asterisk, “*,” next to the bar.



Overdraft usage is also shown in the Type column of RIF output for a Usage Efficiency report. The Type column appears in RIF output only if there are overdraft licenses available or unlicensed usage.

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

4.5 Selecting Output Type and Output File Names

Different output formats are available, depending on the report type selected. Unavailable output formats are grayed out. All reports written to files require an output file name base and at least one output format selection.

The file name base can be entered in one of two methods:

- Type the path and file name base in the text field to the right of the Output File button. If no path is specified, the files are written to the current working directory.
- Click the Output File button to open a file selection browser. Find and double-click the directory in which you want to save the output file. Type the file name base in the file selection browser. Click the Save button.

Output in Text Format

The appropriate file name extension is automatically appended to a file name base, depending on the files' output format:

Table 4-1: File Name Extensions Associated with SAMreport Output Formats

Output Format	File Name Extension
HTML	.html
Text	.txt
Report Interchange Format See a description of RIF in Section 4.9, "Output in Report Interchange Format."	.rif
JPEG/HTML	.jpg/.htm
PNG/HTML	.png/.htm

SAMreport does not warn you if the output file that you are defining already exists and that you are about to overwrite it.

4.6 Output in Text Format

Usage Summary, Raw, and Server reports are available in text format. This format is suitable for printing and reading, but not for parsing.

4.7 Output in HTML Format

Usage Summary and Server reports are available in HTML format. This format is suitable for posting textual reports to a web page.

4.8 Output in JPEG/HTML or PNG/HTML Format

All graphical reports can be generated in JPEG/HTML or PNG/HTML format. Output in JPEG/HTML or PNG/HTML format is intended to be viewed in an HTML browser. PNG/HTML output must be specified in a configuration file or on the command line. Graphical report output in PNG format is of higher quality than JPEG format.

Each page of a graphical report generates two files, a JPEG (PNG) image of the report and an HTML file that wraps the JPEG (PNG) file. Sequential numbers (starting with 0) are appended to the file name base of the JPEG (PNG) files and they have the extension `.jpg`. Sequential numbers (starting with 0) followed by “g” are appended to the HTML file name base and they have the extension `.htm`. A table of contents HTML file is generated for each JPEG/HTML (PNG/HTML) report (called `output_filename_base_toc.htm`).

We recommend that the report output from each JPEG/HTML (PNG/HTML) report be written to its own directory and that JPEG/HTML (PNG/HTML) output be generated in a separate run of the report generator.

4.9 Output in Report Interchange Format

Output in Report Interchange Format (RIF) is available for all reports except Summary Barchart reports. RIF is a simple, flexible, plain text format that is designed to pass tabular information between the report generator and another application, such as a spreadsheet.

The default field separator is a space character, although a different delimiter can be specified using the RIF Delimiter setting on the Advanced pane. Fields that contain the specified field delimiter are surrounded in quotation marks. Do not use double quotes (“”), single quotes (’), or a backslash (\) as the RIF delimiter.

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 file name 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`.

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 unaffected.

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.

Output Arrangement and Detail

All reports show license usage by feature. To create a report that shows more detail about the nature of license usage, you can use a combination of:

- Summary period
- Breakdown category
- Sort order specification
- Feature description

Using these controls, you can decompose the usage of each feature into useful groupings according to your particular reporting requirements.

5.1 Summary Period

The summary period is the period of time used to calculate the summary information. For Usage Over Time and High Water Mark reports, the summary period represents the granularity of the report: one second for a Usage Over Time report and one hour for a High Water Mark report, by default. Longer summary periods are useful when generating Usage Over Time and High Water reports over long reporting periods to control the size of the report. For Summary Barchart and Usage Summary reports, the summary period represents the time interval by which to subdivide the report—an individual bar or line of data is displayed per summary period. For example, using a summary period of Month shows usage statistics on a monthly basis for the entire reporting period.

Note that changing the default summary period setting affects the efficiency of the report generator. Setting a summary period for a Usage Over Time or High Water Mark report reduces the amount of data and can result in less overall memory usage and faster execution and display of a report. Conversely, setting a summary period for a Summary Barchart or Usage Summary report

significantly increases the amount of calculations and memory required. Choose a summary period that is appropriate for the type of report you are generating and the duration of your report data.

The available Summary Period settings are None, Minute, Hour, Day, Week, Semi-Month, and Month. The summary period is set in the main window.

To set a summary period at the command line, see Chapter 8, “Generating Command-Line Reports.”

5.2 Breakdown Category

You can decompose the usage for each feature by selecting one or more breakdown categories. These breakdown categories pertain either to the client application (for example, user, display, or host) or to the license (for example, feat_hostid, the hostid for a node-locked license).

You may use the Arrange pane to specify the breakdown categories. To specify settings in the Arrange pane:

1. Click the Arrange tab in the *SAMreport* main window.
2. Click a breakdown category from the Breakdown Categories Exclude list.
3. Click the << button to move the category from the Exclude list to the Include list.

Or you may use the Summary choice lists in the *SAMreport* main window which provide easy access to the commonly used breakdown categories. Notice that changing the breakdown settings automatically changes the sort order to defaults that correspond to your breakdown settings.

If you specify multiple breakdown categories, the resulting report shows use of features by each unique combination of breakdown categories. For example, if breakdown by user and breakdown by display are both specified, the resulting report shows the usage of each feature by each unique combination of user and display.

Total feature usage is also shown in the entries labeled “All.” Note that the counts for the individual entries may not sum to the counts for the “All” entries. If the combined count for the individual entries for a feature is higher than the count in the “All” entry for the feature, this indicates that some licenses were shared according to the duplicate grouping policy for that feature. See Section 5.5, “Duplicate Grouping,” for additional information about duplicate grouping.

When you specify more than one breakdown category in the Arrange pane, be sure to place the most important category at the top of the list. If you sort the report by breakdown category, the first breakdown category is a primary sort key.

To set breakdown categories at the command line, see Chapter 8, “Generating Command-Line Reports.” To see the list of available breakdown categories, see Section D.2, “Breakdown Categories.”

If you break down usage by group (`GROUP` defined in an end-user options file), you should either:

- 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 in more than one `GROUP`

In the event that a user belongs to more than one group and the groups are included with the report, `SAMreport` arbitrarily chooses one group for reporting purposes. This condition is flagged with a warning.

5.3 Sort Order Specification

Report output can be ordered by feature name and by the breakdown category value. When you select breakdown category settings, default sort order settings are automatically set. You may override these default settings by changing the sort order specifications in the Arrange pane.

Reports containing the same data can be sorted differently. For example, Reports 1 and 2 have breakdown categories of feature and user. Report 1 is ordered by breakdown (user), then feature; Report 2 is ordered by feature, then breakdown (user). The output of each report is shown below:

Report 1:	Report 2:
All	Feature 1
Feature 1	All
Feature 2	User 1
Feature 3	User 2
User 1	Feature 2
Feature 1	All
Feature 2	User 1
Feature 3	

Feature Description

```
User 2                                Feature 3
      Feature 1                        All
      Feature 3                        User 1
                                           User 2
```

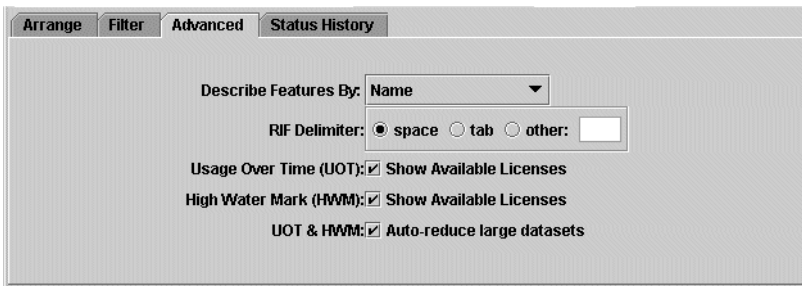
To set sort order specifications at the command line, see Chapter 8, “Generating Command-Line Reports.”

5.4 Feature Description

FLEXnet Licensing license servers automatically combine the count of equivalent FEATURE lines in a license file. This practice is called *pooling*. The collection of licenses which are treated as equivalent are known as a *pool*. FEATURE lines that are not equivalent are counted as distinct pools by the license server, and they can also be reported separately.

By default, *SAMreport* distinguishes features by name only. But if you are interested in seeing usage information for licenses for the same feature that did not pool, you may change the setting in the Advanced pane from Name to:

- Name & Version
- Name & Asset Info
- Name & License Key



For example, if you wanted to distinguish between different versions of each feature, select Name & Version.

If you distinguish features by Name & Asset Info, the feature name and corresponding asset_info value from the FEATURE line is shown on the report. As an end user or license administrator, you can use the asset_info attribute on a FEATURE line to assign a site-specific designation to that

FEATURE line. See the *FLEXlm End Users Guide* for more information about `asset_info`. For example, in the feature line below, the value of `asset_info` is `dept32` to denote a “department 32 asset.”

```
FEATURE FLEXadmin gsi 1.00 20-sep-1996 20 \
SIGN=123456789012 asset_info="dept32"
```

The Name & License Key setting may be useful for analysis, but probably not for presentation, because it includes a FEATURE line’s license key, a 12-20 character string which is unique for the license pool.

To distinguish the usage of different license pools of a feature at the command line, see Chapter 8, “Generating Command-Line Reports.”

5.5 Duplicate Grouping

Duplicate grouping, sometimes known as *license sharing* is a term that describes the practice by the license server of satisfying a license checkout request with a license that is already in use. Duplicate grouping is explicitly controlled for each feature by the vendor, either in the license file or in the FLEXenabled-licensed application itself. A FEATURE/INCREMENT line with duplicate grouping enabled contains “DUP_GROUP=”.

Possible duplicate grouping specifications are:

u	user
h	host
d	display
v	vendor-defined string

For example, duplicate grouping by user results in reuse of a single checked-out license for successive checkout requests for a feature by the same user.

If duplicate grouping is set, and you specify a report that breaks down duplicated license usage, the resulting report shows the same license usage information for each member of the duplicated group. For instance, if duplicate grouping by host is enabled and two users running on the same host simultaneously check out a feature, the requests are grouped and only one license is used. But a report with breakdown by user indicates that one license

was issued to each of the two users. Therefore, breaking down duplicate grouping information into its components or summing individual Raw license usage events can create the impression of excess license utilization.

Several fields can be displayed in a Raw report to help analyze duplicate grouping: `handle`, `shared_handle`, `shared_group`.

5.6 Reporting on Usage by Project

Reports can be created showing the usage of a feature by project. To record usage data by project, each user in the project must set the `LM_PROJECT` environment variable to the project name before starting the client application. This causes the value of `LM_PROJECT` to be written into the report log. Then when generating a report, you can break down or filter data by the `user_project` category.

See the *FLEXnet Licensing End User Guide* for more information about `LM_PROJECT`.

Data Filters

Much of the information contained in the report log is subject to filtering, and any number of filters can be specified for a report. You can specify the particular features, users, hosts, IP addresses, displays, vendor-defined strings, projects, event types (used, denied, etc.), dates, “false” denials, and time periods to include or exclude in you report. Most filters can be set in the Filter pane.

6.1 Defining an Event Filter

Multiple filter categories for one report are applied with a Boolean AND operation—all the specified conditions must be satisfied. If a report is filtered on `user==User1` and `feature==Feature2`, the resulting report contains usage of Feature2 by User1.

Multiple instances of a single filter category are applied with a Boolean OR operation. If a report is filtered on `user==User1` and `user==User2`, the resulting report contains usage of features by either User1 or User2.

You can set event filters from the complete list of filters in the Filter pane. Multiple filter values for the same filter category can be specified on the same line separated by a space or on separate lines. For example, `user==User1 User2` is equivalent to `user==User1` and `user==User2`. Entries for feature and user filters in the Filter pane do not appear in the Feature and User text boxes in the *SAMreport* main window, but they are still applied to the report.

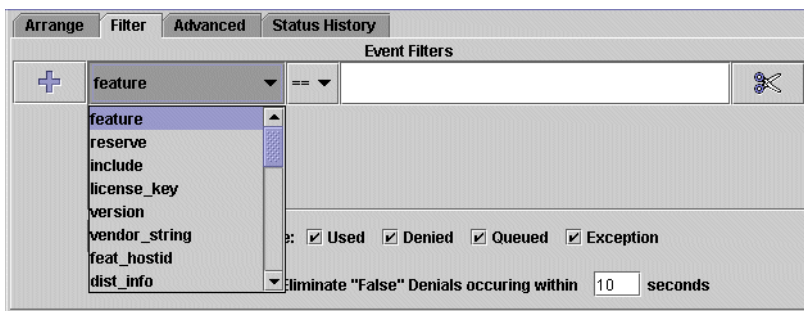
Defining an Event Filter

Table 6-1 lists valid operators in SAMreport filters:

Table 6-1: Valid Operators in SAMreport Filters

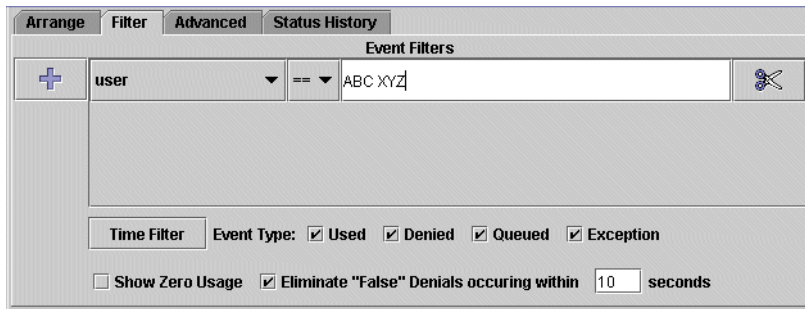
Operator	Description
==	Identical to
~=	Includes (with wildcards) * Match any number of characters. ? Match any single character. <i>pattern</i> * Match <i>pattern</i> at beginning of text. * <i>pattern</i> Match <i>pattern</i> at end of text. * <i>pattern</i> * Match <i>pattern</i> anywhere in text. * <i>pattern1</i> * * <i>pattern2</i> * To match multiple patterns, separate with spaces.
!=	Not equal to
!~	Excludes (with wildcards) See discussion for =~.
>	Greater than
<	Less than

1. Click the Filter tab in the SAMreport window.
2. Click the Add (+) button. Click feature to open the Field Name choice list.

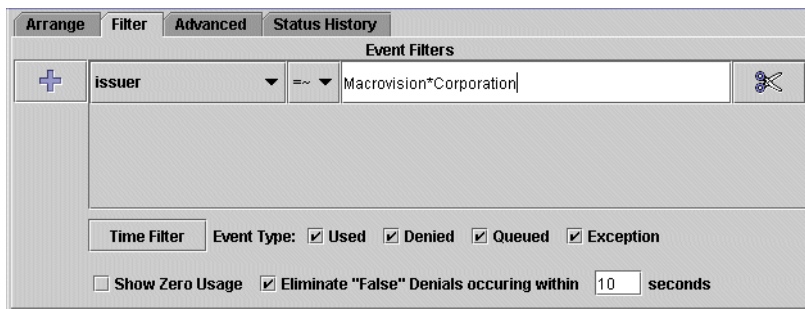


3. Select a field name. Then click == to open the Operator choice list. Select an operator.

- Click in the Value field of the entry and type the filter value. Separate multiple entries with a space.



To match a value that contains spaces, use the =~ operator and replace the spaces with asterisks.



To add another filter, click the Add (+) button again. To remove an individual filter from the Filter pane, click the Cut (scissors) button.

As a shortcut, you can specify one or more feature and user filter categories in text boxes in the middle portion of the *SAMreport* window. Separate multiple entries with a space. Entries that you make in these text boxes do not get translated to the Filter pane, but they are applied when the report is generated.

Defining a Time Period Filter

Report Type: Usage Summary
Summarize: By User Then Feature
Summary Period: None
Feature: FLEXadmin GLOBEtrack
User: admin
Start Date:
End Date:
Output File: C:\MVSN\examples\samreport\output\test
 HTML Text RIF JPEG/HTML

To set data filters at the command line, see Chapter 8, “Generating Command-Line Reports.” A complete set of available filter categories is available in Section D.3, “Filter Categories.”

6.2 Defining a Time Period Filter

You can set a period of time during each week on which to filter report data in a Summary report. Only license activity during the specified block of time is included in the report. In the Filter pane, click the Time Filter button to open the Timeperiod Filter Editor.

Mond..	Tues..	Wed..	Thur..	Friday	Satur..	Sund..
07:30	07:30	07:30	07:30	07:30	07:30	07:30
08:00	08:00	08:00	08:00	08:00	08:00	08:00
08:30	08:30	08:30	08:30	08:30	08:30	08:30
09:00	09:00	09:00	09:00	09:00	09:00	09:00
09:30	09:30	09:30	09:30	09:30	09:30	09:30
10:00	10:00	10:00	10:00	10:00	10:00	10:00
10:30	10:30	10:30	10:30	10:30	10:30	10:30
11:00	11:00	11:00	11:00	11:00	11:00	11:00
11:30	11:30	11:30	11:30	11:30	11:30	11:30
12:00	12:00	12:00	12:00	12:00	12:00	12:00
12:30	12:30	12:30	12:30	12:30	12:30	12:30
13:00	13:00	13:00	13:00	13:00	13:00	13:00
13:30	13:30	13:30	13:30	13:30	13:30	13:30
14:00	14:00	14:00	14:00	14:00	14:00	14:00
14:30	14:30	14:30	14:30	14:30	14:30	14:30
15:00	15:00	15:00	15:00	15:00	15:00	15:00
15:30	15:30	15:30	15:30	15:30	15:30	15:30
16:00	16:00	16:00	16:00	16:00	16:00	16:00
16:30	16:30	16:30	16:30	16:30	16:30	16:30
17:00	17:00	17:00	17:00	17:00	17:00	17:00
17:30	17:30	17:30	17:30	17:30	17:30	17:30
18:00	18:00	18:00	18:00	18:00	18:00	18:00

Save & Exit Clear Cancel

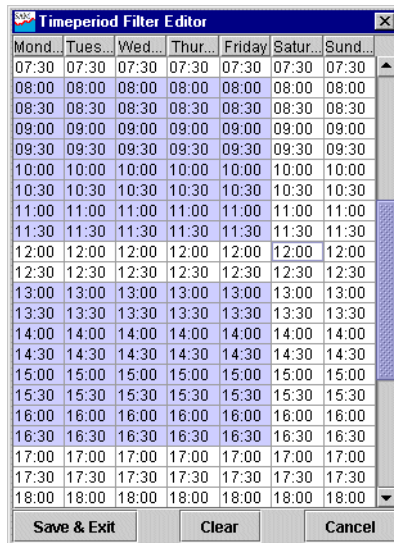
In this form, you can define a block of time to include in the report output. Time is measured with a 24-hour clock. Each block represents a half-hour increment of time, starting with the time on the block label. So the hour-long block of time from 3pm to 4pm includes the 15:00 and 15:30 blocks.

To select a block of time, sweep out the block with the left mouse button. To set this block of time, click the **Save & Exit** button. To clear the form, click the **Clear** button. To remove a selected block of time, hold down the **Ctrl** key and sweep out the block of time.

For example, to select the hours of 8am to noon and 1pm to 5pm, Monday through Friday, you must perform two steps:

1. With the left mouse button depressed, sweep out the block of time from 08:00 to 16:30 from Monday through *Saturday*, inclusive.
2. Hold down the **Ctrl** key, and with the left mouse button depressed, sweep out the 12:00 and 12:30 blocks on *Saturday*.

Note that with the second sweep, you simultaneously removed all hours on Saturday and the noon hour from every weekday.



If you change report type and you want to apply the same time period filtering that you used for the previous report type, you must reset the time period filter.

6.3 Reports 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 compensating for time differences. Reports, on the other hand, show time using the time zone set on the machine where the report was generated. *SAMreport* translates times in UTC from the report log using the offset for the time zone of the machine on which *SAMreport* is running.

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

Possible causes for the times in reports that are different than you expect are:

- 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 *SAMreport* is running is set incorrectly. Even if the clock time looks correct, the time zone may not be set correctly.

6.4 Selecting an Event Type Filter

In the *SAMreport* interface, you can select what type of events recorded in a report log to include in a report — Used, Queued, Denied, or Exceptions (Unsupported). By default, all four of these event types are included in reports.

Graphical reports display only Used events; by default, Usage Summary and Raw reports display all four event types. When reported, the total number of license requests is Used + Denied + Exceptions. Queued events are represented in the total as Denied and/or Used events — a Queued event is initially logged as a Denied event, and if granted, is then logged as a Used event.

6.4.1 Used Events

A Used event consists of a license checkout/checkin pair. The *number* of Used events is summed over the time period of a report, and the *duration* licenses are used is displayed in a report. License usage is calculated as the time between a checkout and its checkin.

6.4.2 Queued Events

A Queued event consists of a queue/dequeue pair. The *number* of Queued events is summed over the time period of a report, but the *duration* a license request is queued before being denied or granted is not displayed (however, queued duration can be calculated from data in a Raw report). If an application is configured for queuing and none of its licenses are available, a Denied event and a Queued event are written into the report log with the same timestamp. The application will continue to try to gain a license until one becomes available. Additional denials are not written for each try while queued or if a license is never granted. A queued request will remain queued until it is granted unless the daemon crashes, the license server is shut down, or the request is cancelled or timed out (usually about 2.5 hours).

QUEUED AND GRANTED

- Request a license when none available, and application is configured to queue
- Log denial due to “Users are queued for this feature” (increment Denied events)
- Log queue (increment Queued events) with same timestamp as denial
- Log dequeue (output from Raw report can be used to calculate queue duration)
- Log checkout (increment Used events and start accumulating usage)
- Log checkin (stop accumulating usage)

QUEUED AND NOT GRANTED

- Request a license when none available, and application is configured to queue
- Log denial due to “Users are queued for this feature” (increment Denied events)
- Log queue (increment Queued events) with same timestamp as denial
- Log dequeue (Raw report can be used to calculate queue duration)

6.4.3 Denied Events

The *number* of denial events is summed over the time period of a report. A denial is instantaneous and has no duration associated with it. The *why* field in a Raw report gives the reason why a denial occurred (see Section 3.7.2, “Displaying Denials in Raw Reports”).

If filtering “false” denials is enabled, the report includes the number and percentage of “false” denials that were eliminated (see Section 6.5, “Eliminating “False” Denials”).

6.4.4 Exception Events

Exception (unsupported) events are generated when the license server receives a request for a feature it does not know about. Exceptions usually occur when the application and the license server are looking at two different license files. The application reads its license file and thinks it can get a license for the feature, but when it asks the license server for a license, the license server denies the request and logs it as an exception.

The most common instances of exceptions are:

- Feature exists in the application’s license file, but not in the license server’s license file.
- Version of the feature in the application’s license file is later than the version(s) in the license server’s license file. (The license server gives out a license if the request is for an earlier version but it does not give out a license if the request is for a later version.)

6.5 Eliminating “False” Denials

If an application requests a license from a license server that uses a license file containing several INCREMENT lines for the same feature, it is possible to get a cluster of denials reported before a successful checkout is obtained. You can eliminate extraneous or “false” denial events by selecting the Eliminate “False”

Denials check box and specifying a time interval in seconds. With this filter set, denials for the same feature, user, host, process id, and report log file that occur in clusters within the specified time interval are considered qualified for elimination. If not specified, the time interval defaults to 10 seconds.

SAMreport analyzes the qualified denials in each report log according to the following criteria and eliminates extraneous denials from a report:

- If the FLEXenabled application is built with *FLEXlm* v9.0 (called FLEXnet Licensing from v10.0 on) or greater:
 - An ultimate denial event is never removed.
 - Regular denials are always removed.
- If the FLEXenabled application is built with a version of *FLEXlm* less than v9.0:
 - All qualified denials occurring within the time interval are considered together. If at the end of the time interval, the license has not been granted, the last denial is reported and all previous denials in the interval are eliminated from the report. If a series of denials is seen followed by a checkout, the checkout is reported and the denials are eliminated.

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

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

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

6.6 Showing Zero Usage

You can select the Show Zero Usage check box to display all features in a report output, regardless of whether they have been used during the report period (for Usage Over Time, High Water Mark, Summary Barchart, and Usage Summary Reports).

6.7 Setting Start and End Dates for a Report

Start and end dates for report output can be set in text fields in the middle portion of the SAMreport main window. Times can be included. If no start and end dates are set, the report is generated with data from the entire report log. If no time is specified, days begin at midnight, 00:00:00.

The following formats are accepted for 3:30pm on January 31, 1999:

MM/dd/yyyy HH:mm:ss	1/31/1999 15:30:00
MM/dd/yyyy HH:mm	1/31/1999 15:30
MM/dd/yyyy	1/31/1999
yyyymmdd HHmmss	19990131 153000
yyyymmdd HHmm	19990131 1530
yyyymmdd	19990131
MMM dd, yyyy HH:mm:ss	Jan 31, 1999 15:30:00
MMM dd, yyyy HH:mm	Jan 31, 1999 15:30
MMM dd, yyyy	Jan 31, 1999

HH is measured by a 24-hour clock (0-23). MMM can be either the three-letter abbreviation for the name of a month or the full name of the month (Jan, January, jan, january, JAN, JANUARY).

Configuration Files

Saving a report configuration is useful if a report is generated repeatedly and only a few of its parameters change from run to run. For example, input report log file names and start and end dates may change every time a report is generated, but other settings remain the same.

When you run *SAMreport* interactively, the File | Save Configuration selection allows you to save all the current settings to a file. You can also build a report configuration file in a text editor. If you plan to use the configuration file from the command line, the file path must not contain any spaces. See Appendix D, “Command-Line Report Reference,” for configuration file syntax.

A report configuration file can be used to load report settings in *SAMreport* by selecting File | Load Configuration or by specifying the file when running from the command line.

The following configuration file generates a High Water Mark report for the features FLEXadmin and GLOBEtrack and write output to RIF files.

```
-name "High Water Mark"
-descrip "High Water Mark Graphic Report"
-report_type summary
-input "C:\Program Files\Macrovision\samreport\version\
      machind\samples\sample.rl"
-filter feature == "FLEXadmin GLOBEtrack"
-uot
-rif
-output C:\MVSN\examples\samreport\output\uot_test
-feat_uniq name
-interval Hour
```

The following configuration file generates a Usage Summary report that shows each user and the feature used. The report describes features by feature name and version, summarizes data over intervals of a month, and includes data from 9am to 5pm Monday—Friday, January 1, 1996 to March 31, 1996. It writes one output file called `sample.txt` and one called `sample.html`.

```
-name "Usage Summary"-descrip "Usage Summary Text Report,  
available in plain text, HTML, and RIF"  
-report_type summary  
-input "C:\Program Files\Macrovision\samreport\version  
machind\samples\sample.rl"  
-output C:\MVSN\examples\samreport\output\sample  
-breakdown user  
-filterperiod "2.9:0:0" "2.16:59:59"  
-filterperiod "3.9:0:0" "3.16:59:59"  
-filterperiod "4.9:0:0" "4.16:59:59"  
-filterperiod "5.9:0:0" "5.16:59:59"  
-filterperiod "6.9:0:0" "6.16:59:59"  
-html  
-text  
-usage_summary  
-feat_uniq name_ver  
-orderby bf  
-startdate "01/01/1996 00:00:00"  
-enddate "03/31/1996 00:00:00"  
-interval Month
```

A sample configuration file is included in the *SAMreport* installation at *SAMreport_home/machind/samples/uotg.config*. This configuration file contains some sample display options for a Usage Over Time report that cannot be set in the *SAMreport* GUI.

Generating Command-Line Reports

There are several reasons why you might want to generate a report from the command line instead of generating one interactively through the *SAMreport* window:

- There are more options available from the command line.
- You may want to generate the same report periodically with many of the same parameters.
- You may want to automate generating reports.

8.1 Starting with the *SAMreport* GUI

The simplest way to design a batch report configuration is to use the *SAMreport* interface to configure and save the settings you would like to use. You can select File | Save Configuration to save your report configuration to a file. If you plan to use the configuration file from the command line, the file path must not contain any spaces. Save the configuration file to a location whose path does not contain spaces.

The report configuration file is a plain text file. You may use a text editor to view and edit the report configuration. For example, the fields of Raw reports cannot be saved into a configuration file through the *SAMreport* interface. To define Raw display fields in a configuration file, they must be typed into the file.

Note: Do not specify report and output file names in the configuration file if you want the names to be different each time you generate the report; it is best to specify these file names on the command line.

8.2 Starting with the Command Line

You may configure an entire report on the command line, without the use of configuration files. 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	<code>report Summary -uot</code>	RIF JPEG/HTML PNG/HTML
	High Water Mark	<code>report Summary -uot -interval period</code>	RIF JPEG/HTML PNG/HTML
	Summary Barchart	<code>report Summary -barchart metric</code>	JPEG/HTML PNG/HTML
	Usage Efficiency	<code>report Summary -usage_efficiency -barchart metric</code>	RIF JPEG/HTML PNG/HTML
	Usage Summary	<code>report Summary -usage_summary</code>	HTML Text RIF
Server	Server Coverage	<code>report Server -coverage</code>	Text
Raw	Raw	<code>report Raw</code>	Text RIF

On Windows, a complete list of command-line arguments, breakdown categories, and filters for Summary, Raw, or Server reports can be written to a file, `samlog.txt`, by typing one of the following commands in an MS-DOS Command Prompt window at `SAMreport_home`.

On UNIX, a complete list of command-line arguments, breakdown categories, and filters for Summary, Raw, or Server reports can be displayed by typing one of the following commands at the command line in the *SAMreport_home* directory.

Table 8-2: Commands to Invoke SAMreport Help

Report Type	Command
Summary	<code>report Summary -help</code>
Raw	<code>report Raw -help</code>
Server	<code>report Server -help</code>

Options to the Java launcher are listed in Appendix C, “Launcher and Java Options.” Command-line arguments for `report` current as of this writing are described in detail in Appendix D, “Command-Line Report Reference.”

8.3 Generating a Report from the Command-Line

To generate a report from the command line:

1. At a command line, change to the *SAMreport_home* directory.
2. Type:

```
report report_family -input input_filename1
    [input_filename2...] output_format [-output
    output_filename_base]
    [optional_report_args...]
    [-- optional_launcher_args...]
```

For example:

```
report Summary -input input_file1 input_file2
    -output output_file -config myreport.config
```

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

8.4 Specifying a Different License File

If you want to use a *SAMreport* license file different from the one located in *SAMreport_home/machind/license/license.dat*, it can be specified when running a report from the command line:

```
report report_family -c license_file_path report_args
```

8.5 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 *SAMreport*. If you want to use more than one report log, enter the path to each report log separately. If you use *-input*, 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 in Table 8-3:

Table 8-3: Output Formats and Their Command-Line Arguments

Output Format	Command-Line Argument
HTML file	-html
Text file	-text
Report Interchange Format files	-rif
JPEG/HTML files	-jpeg
PNG/HTML files	-png

8.6 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 ...]
```

Setting Breakdown Masks

A complete set of available breakdown categories is available Section D.2, “Breakdown Categories.”

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}. See Section 5.4, “Feature Description,” for the definitions of these ids.

8.7 Setting Breakdown Masks

A breakdown mask is a feature available in *SAMreport* only at the command line. Using the mask, you can specify which characters of the breakdown value are to be used to group data in a report. This masking technique is useful for reporting on subnets.

The mask is position-dependent. It is a character-by-character mask that identifies which characters are used and which characters are ignored. If non-consecutive characters of the breakdown value are specified by the mask to be used, the ignored characters are replaced with a ‘-’ for better readability. If the breakdown value is longer than the mask, the tail of the breakdown value is ignored.

Syntax:

```
-breakdown breakdown_field[/[OX][OX]*]
```

Example:

```
-breakdown user_ipaddr/0000000000XXXX
```

This example masks out the final triplet of digits (including the dot) in each IP address. The first three triplets of the IP address are used to group data, thereby grouping events from the same subnet. For example, events from 192.171.244.020 and 192.171.244.034 would be grouped together and reported as events from 192.171.244. Events from 192.171.230.031 would be reported as events from 192.171.230.

8.8 Aliasing Feature or User Names in Report Output

By default, feature and user names appear in report output as they appear in the license file and report log. If you would like to display clearer or more meaningful feature or user names in your report output, the actual user and feature names can be replaced in the report output by specifying alias values for them. If you define a filter on a feature or user value for which an alias has been defined, specify the alias in the filter value.

```
-alias type from to
```

where:

<i>type</i>	feature or user.
<i>from</i>	Value in report log.
<i>to</i>	Alias to appear in report output. If the alias contains spaces, enclose the alias in double quotes.

For example:

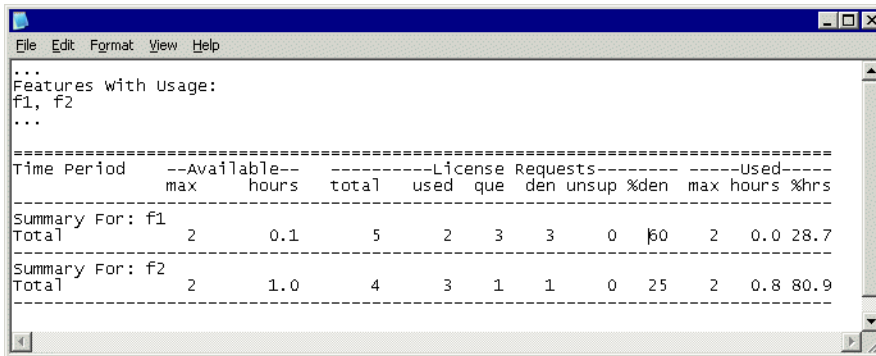
```
-alias user rander "Ruth Anderson"
-alias user dworle "Debbie Worley"

-alias feature f1 "Easy Editor"
-alias feature f3 "Easy Calculator"
```

It is possible to map more than one user name to the same alias. If you have v5+ report logs, it is also possible to map more than feature name to the same alias.

Aliasing Feature or User Names in Report Output

The following sample shows report output for features f1 and f2 before mapping both to a single alias.

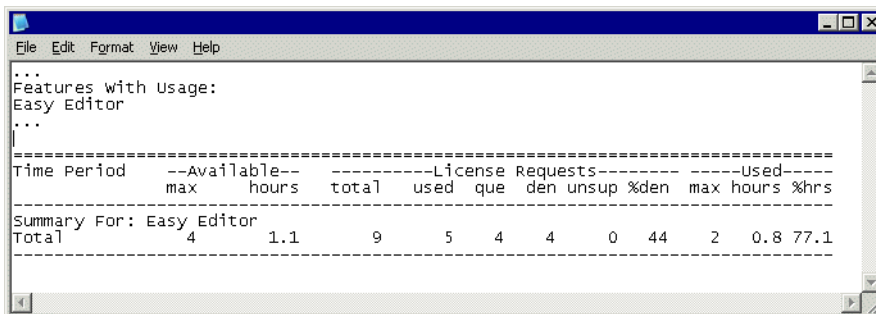


```
File Edit Format View Help
...
Features with Usage:
f1, f2
...
-----
Time Period  --Available--  -----License Requests-----  --Used--
             max    hours  total  used  que  den  unsup  %den  max  hours  %hrs
-----
Summary For: f1
Total        2        0.1    5     2   3   3     0   60    2    0.0  28.7
-----
Summary For: f2
Total        2        1.0    4     3   1   1     0   25    2    0.8  80.9
-----
```

The following sample shows report output after both features f1 and f2 were aliased to Easy Editor.

```
-alias feature f1 "Easy Editor"
-alias feature f2 "Easy Editor"
```

Because f1 and f2 are aliased to the same name, the number of licenses available, used, denied, etc. and the hours used and available for the two individual features are summed and shown on the report for the pseudo-feature Easy Editor.



```
File Edit Format View Help
...
Features with Usage:
Easy Editor
...
|
-----
Time Period  --Available--  -----License Requests-----  --Used--
             max    hours  total  used  que  den  unsup  %den  max  hours  %hrs
-----
Summary For: Easy Editor
Total        4        1.1    9     5   4   4     0   44    2    0.8  77.1
-----
```


8.9 Defining Field Filters

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

```
-filter category operator value
```

where:

<i>category</i>	Filter category.
<i>operator</i>	Relational operator.
<i>value</i>	Value for comparison. If an alias has been specified for a feature or user value, the alias must be specified as the filter value.

FLEXnet Licensing allows spaces in the values of some category types, but matching against values containing spaces is not supported in *SAMreport*. A possible workaround is to use the `=~` operator and replace spaces with asterisks. For example:

```
issuer =~ "Macrovision*Corporation"
```

Table 8-4 lists valid operators in *SAMreport* filters:

Table 8-4: Valid Operators in *SAMreport* Filters

Operator	Description
<code>==</code>	Identical to
<code>=~</code>	Includes (with wildcards) * Match any number of characters. ? Match any single character. <i>pattern</i> * Match <i>pattern</i> at beginning of text. * <i>pattern</i> Match <i>pattern</i> at end of text. * <i>pattern</i> * Match <i>pattern</i> anywhere in text. * <i>pattern1</i> * * <i>pattern2</i> * To match multiple patterns, separate with spaces.
<code>!=</code>	Not equal to

Table 8-4: Valid Operators in SAMreport Filters

Operator	Description
!~	Excludes (with wildcards) See discussion for =~.
>	Greater than
<	Less than

A complete set of available event filter categories is available in Section D.3, “Filter Categories.”

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
```

Date formats can be found in Section 6.7, “Setting Start and End Dates for a Report.” 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, *SAMreport* 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.

8.10 Customizing Appearance of Graphical Report Output

There are several options that allow you to customize the text style and colors used in graphical report output. When a color is specified, it can be one of the following set of colors:

```
black          cyan          magenta       white
blue          green         red           yellow
```

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
```

Customizing Appearance of Graphical Report Output

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

Table 8-5: Font Settings for Graphical Reports

Option	Description	Constraints
<code>-fontface</code> <i>fontname</i>	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</code> <i>points</i>	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]*
```

- 1 Solid line. If datastyle 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:

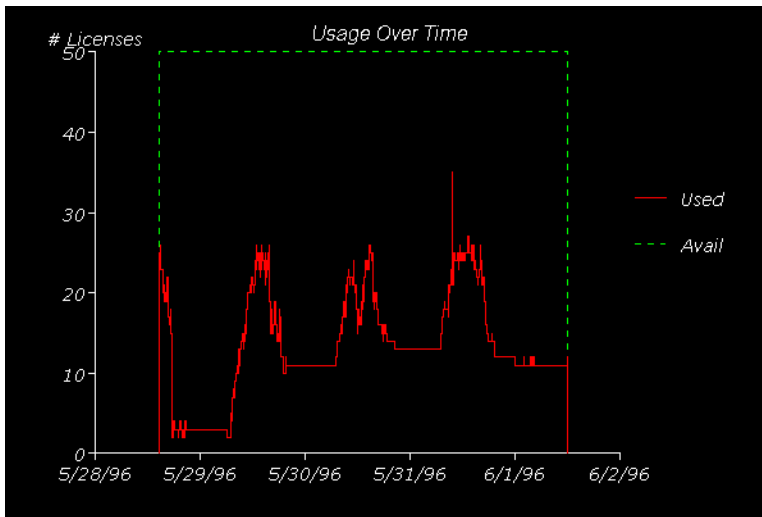
```
-linewidth pixels
```

Consider the following settings:

```
-input machind\samples\sample.rl
-output test
-uot
-png
-title "Usage Over Time"
-fontsize 14
-fontface "Verdana"
-fontitalic
-bgcolor black
-textcolor white
-datacolor red,green
-datastyle -1,0
```

Note that a consequence of specifying a title is the loss of the feature name in the output.

The resulting output looks like:



8.11 Loading a Configuration File

When generating a report from the command line, *SAMreport* obtains report configuration information from the command line explicitly, from a plain text configuration file, or both. Command-line arguments read directly from the configuration file are indistinguishable from arguments found on the command line. If conflicting values for an argument are specified, the last value specified is applied. Configuration files can be nested, if desired, 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:

```
-config config_filename
```

8.12 Defining a Raw Report

If you generate a Raw report from the command line, you can choose which fields to display. The default set of Raw report fields are defined in and read from *SAMreport_home*\machind\samples\raw.config. Fields are displayed in the order that you specify them. Each field specification consists of the `-field` argument, the field name, the display width of the field, and the field justification:

```
-field field_name,field_width,justification
```

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
```

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

You can specify `-field why` to display why each license request was denied. For a complete listing of available fields in Raw reports, see Section D.4, “Raw Report Fields.”

Defining a Raw Report

Enabling and Managing Report Logs

In order to use *SAMreport*, you need one or more report log files that are locally accessible (via NFS) to *SAMreport*. Each report log must be written by a single vendor daemon process on a single host. For example, if *vdaemon1* is running on three different license server hosts, each copy of *vdaemon1* should write a report log to the host on which it is running.

As a vendor daemon runs for a period of time, report log output can become quite large. Over time it may be necessary to move report log output into different files that contain license activity over a specified period of time.

Note: *SAMreport* **cannot** produce reports from debug log files (*lmgrd*'s standard output).

A.1 Finding a Report Log

You can find out if you have a report log for a specific vendor daemon by checking the DAEMON line in the vendor's license file. If there is a path to an options file at the end of the DAEMON line, check that options file for a path to a report log. For example:

WINDOWS

If the license file contains the following DAEMON line:

```
DAEMON hij C:\flexlm\hij C:\flexlm\hij.opt 4090
```

you would look for the options file at `C:\flexlm\hij.opt`. If, in the options file, you saw the following line:

```
REPORTLOG +C:\flexlm\hij.r1
```

you would look for your report log file at `C:\flexlm\hij.r1`

Identifying a Report Log

UNIX

If the license file contains the following DAEMON line:

```
DAEMON hij /usr/local/hij /usr/local/hij.opt 4090
```

you would look for the options file at `/usr/local/hij.opt`. If, in the options file, you saw the following line:

```
REPORTLOG +/usr/local/hij.rl
```

you would look for your report log file at `/usr/local/hij.rl`

If the DAEMON line in your license file does not specify a path to an options file or if your options file does not have a REPORTLOG line, then you probably don't have a report log for this vendor daemon. The exception to this is if the FLEXnet Licensing utility, `lmswitchr`, has been run on the license server. See below for details.

SAMSUITE

If you have imported your license files into *SAMsuite*, installed your folders, and restarted your license servers with *SAMsuite*, and you accepted the default setting to enable report logging, then you have report logs for your vendor daemons.

If you have not restarted your license servers with *SAMsuite*, you may or may not have report log files.

A.2 Identifying a Report Log

FLEXlm v5+ (called FLEXnet Licensing from v10.0 on) report logs start with:

```
# FLEXlm Report Log, ...
```

FLEXlm v5+ (called FLEXnet Licensing from v10.0 on) report logs are compressed and authenticated and look similar to uuencoded files.

FLEXlm pre-v5 report log files always start with:

```
lm_log_start
```

and contain lines such as:

```
lm_timestamp 822934842 "cdslmd"  
lm_feature 822934842 "cdslmd" "300" 4.3 1 0  
lm_usage2 822934985 "cdslmd" 0X73858 0X0 0 CREQ "300" 4.3 1 1  
"george" "abc" ":0.0" "Hidden" 0 62619
```

A.3 Turning on a Report Log

SAMreport can read report logs that are locally accessible to the machine where *SAMreport* runs—either by being located on the machine where *SAMreport* runs or being located on a disk that is accessible via NFS. You can configure a license server to write to an NFS-mounted directory local to the license server machine, then export that directory for the machine where *SAMreport* runs to have access to report logs.

After you enable report logging with one of the following methods, your vendor daemon writes a report log from which you can generate a report with *SAMreport*. Depending on your license usage, you should wait a few hours or a few days before generating a report from this report log.

The options files and license files for each license server in a redundant license server configuration must also be set up in order for report logging to continue when the master changes. If a master license server goes down, the new master reads its license and options files and writes to its own report log.

Note: For v8.0+ vendor daemons, rereading a license file also causes new lines in the options file to be read. When you change or create an options file for a vendor daemon older than v8.0, you must restart the license server in order for those changes to take effect. See Section A.7.1, “Finding Vendor Daemon and FLEXnet Licensing Utilities Versions.”

WINDOWS

To enable report logging for one of your FLEXnet Licensing vendor daemons:

1. Create an options file for the vendor daemon, if one doesn't already exist. The options file is a plain text file with any name (we recommend an `.opt` extension).

For example, for a vendor daemon called `vend`, the options file could be:

```
C:\MVSN\options\vend.opt
```

2. In the options file, enter a line of the following form:

```
REPORTLOG +path_to_report_log
```

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

We recommend naming the report log `vendor.r1`.

Turning on a Report Log

For example, for a vendor daemon called `vend`, you could enter:

```
REPORTLOG +C:\MVSN\logs\vend.rl
```

Each vendor daemon must write its own report log. Do not direct multiple vendor daemons to write to the same report log. *SAMreport* (starting with v3.0d) does not process any report log that has been written by more than one vendor daemon on more than one host.

3. Add the path and file name of the options file to the DAEMON line in the license file. For example:

```
DAEMON vendor [vendor_daemon_path]\  
          [[options=]options_file_path] [[port=]port]
```

4. Depending on the version of your vendor daemon, reread the license file or restart the license server.

UNIX

If your vendor daemons are not writing report logs, there are three ways to get a report log started. The method you choose depends upon whether you want to restart your license server and whether you want to use *SAMsuite* to restart it.

METHOD 1: USE SAMSUITE TO RESTART LICENSE SERVER WITH REPORT LOGGING ENABLED

Refer to *SAMsuite* documentation for instructions on importing license files, installing folders, and restarting license servers. Accepting the default settings during those processes turns on report logging.

When you enable report logging and restart your license server with *SAMsuite*, the vendor daemon writes the report log to the Folder Install Directory. The report log file is named `vendor.rl`.

If you already imported your license files but turned off report logging, you can enable logging, install each folder, restart each license server as follows:

1. Select Reports | Enable Logging. You see a list of your folders.
2. Select a folder for which you wish to enable reporting. A multichoice list of vendors contained in the selected folder are displayed.
3. Vendors with report logging enabled are highlighted. Select (highlight) any additional vendors for which you wish to enable reporting.
4. Press OK at the bottom of the form. Pressing the Cancel button causes any changes to be discarded.
5. Install the folder (Folder | Install) and, depending on the version of your vendor daemon, restart the license server (Control | Restart).

METHOD 2: RESTART LICENSE SERVER OUTSIDE SAMSUITE AFTER CREATING AN OPTIONS FILE AND SPECIFYING PATH TO OPTIONS FILE IN LICENSE FILE

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

```
REPORTLOG +path_to_report_log
```

The options file is a plain text file with any name.

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

We recommend calling the report log *vendor.rl*.

Each vendor daemon must write its own report log. Do not direct multiple vendor daemons to write to the same report log. *SAMreport* (starting with v3.0d) does not process any report log that has been written by more than one vendor daemon on more than one host.

2. Add the path to your options file to the DAEMON line in the license file. For example:

```
DAEMON vendor [vendor_daemon_path]\n\n  [[options=options_file_path] [[port=port]
```

3. Depending on the version of your vendor daemon, reread the license file or restart the license server.

For example, to make vendor “vend” write a report log to */usr/app/logs/vend.rl*, you could do the following:

1. Create an options file named

```
/usr/app/license/vend.opt
```

containing one line:

```
REPORTLOG +/usr/app/logs/vend.rl
```

2. Edit the license file, adding the path to the options file to the vend DAEMON line. So the DAEMON line might look like this:

```
DAEMON vend /usr/app/bin/vendor_daemon\n\n  /usr/app/license/vend.opt
```

3. Reread the license file or restart the license server.

METHOD 3: USE THE FLEXNET LICENSING LMSWITCHR UTILITY

Note: `lmswitchr` does not work with FLEX`lm` v3.0 or v4.0 vendor daemons.

Issue the following FLEXnet Licensing command:

```
lmswitchr -c path_to_lic_file feature_name  
           path_to_report_log
```

We recommend calling the report log `vendor.rl`.

Each vendor daemon must write its own report log. Do not direct multiple vendors to write to the same report log. `SAMreport` (starting with v3.0d) does not process any report log that has been written by more than one vendor daemon on more than one host.

If you don't have `lmswitchr`, but you have any of the other FLEXnet Licensing utilities, such as `lmreread` or `lmdown` or `lmutil`, copy one to `lmswitchr`. These are all really just one binary whose behavior is determined by its name.

Be sure to specify the path to the license file that is currently in use by the license server. `feature_name` is any feature managed by the vendor daemon for which you are starting a report log. You name one feature and it turns on the report log for all features for that vendor daemon. With v5.1 or later of `lmswitchr`, you can name the vendor daemon instead of one feature. Check the version of `lmswitchr` by typing:

```
lmutil lmver lmswitchr
```

You should also do steps 1 and 2 under Method 2 above. This way, if the license server goes down and comes back up, the vendor daemon continues to append data to the report log.

A.4 The Size of the Report Log

Because of a report log's format, standard file compression tools can significantly reduce the size of a report log file when it is no longer being written to. To avoid corruption and for performance, we suggest that each vendor daemon write to its own report log file on a disk that is local to the machine that is running the vendor daemon.

A good estimate of the size of your report logs (from FLEX lm pre-v5 vendor daemons) is 100 bytes per checkout or checkin, or about 20-25% bigger than the portion of the debug log pertaining to that vendor daemon. FLEX lm (called FLEXnet Licensing from v10.0 on) v5+ daemons generate report log files that are approximately 25% of the size of pre-v5 report log files.

UNIX

If you have a debug log file which is written by more than one vendor daemon, you could `grep` the debug log for the daemon name, pipe it through `wc -c` for the size, then multiply by 1.25. This would give a rough approximation of the size of the report log after the same amount of time as the period covered in the debug log file.

A.5 Flushing Data to a Report Log

For efficiency, the license server periodically flushes usage events to the report log. The frequency of these flushes depends on the activity level of the license server as well as on the version and configuration of the vendor daemon. To generate a report with up-to-the-minute usage information, before generating the report, manually flush data from the buffer to the report log by causing the vendor daemon to reread its license file with the FLEXnet Licensing utility, `lmreread`.

WINDOWS

On Windows, you can issue an `lmreread` command by clicking the Reread License File button in the Stop/Reread tab of LMTOOLS.

UNIX

On UNIX, you can issue `lmreread` through SAM $suite$ by selecting Control | Reread or at the command line by typing:

```
lmreread -c path_to_lic_file [-vendor_daemon]
```

A.6 ssreport: A Report Log Utility

You can view a brief description of a report log by using the `ssreport` utility included with SAM $report$. The `ssreport` utility is a standalone component that reads a report log. It shows the size of a report log, start and end dates, duration, and vendor daemon name and version.

On UNIX, `ssreport` is in the `SAMreport_home/platform` directory. On Windows, `ssreport.exe` is in the `SAMreport_home/bin` directory.

Managing Report Logs

Usage:

```
ssreport [[-o out_file] | [-a out_file]] report_log
```

where:

- o Specify the output file name
- a Append to the specified output file

A.7 Managing Report Logs

Remember that you can produce a report from multiple report logs as long as each report log is written by a single vendor daemon running on a single machine (vendor daemons running on more than one redundant server must write to their own report logs). Therefore, to keep the size of report logs manageable, you should regularly close out a current report log and begin to write to a new one. This process is known as rotating report logs. Later in this section are recommended methods for rotating report logs.

You may want to automate the process of managing each license server's report logs. The following steps can be performed on a regular basis:

WINDOWS

1. Rotate a report log. To avoid corruption in your report logs, we recommend that you rotate your logs with the FLEXnet Licensing utility, `lmswitchr`, found in `LMTOOLS`.
2. Update the vendor daemon's options file(s).
3. Move the older report log to a designated location.
4. Run `ssreport` to create a description of the report log (see Section A.6, "ssreport: A Report Log Utility").
5. Archive (and probably compress) the report log and its description when it is no longer needed to generate reports.

UNIX

1. Rotate a report log. To avoid corruption in your report logs, we recommend that you rotate your logs with either `lmswitchr` or `lmnewlog`.
2. If using `lmswitchr`, update the vendor daemon's options file(s).
3. Move the older report log to a designated location.
4. Run `ssreport` to create a description of the report log (see Section A.6, "ssreport: A Report Log Utility").
5. Archive (and probably compress) the report log and its description when it is no longer needed to generate reports.

SAMSUITE

1. Rotate a report log. To avoid corruption in your report logs, we recommend that you rotate your report logs with the *SAMsuite* batch option, `-switchlog`.
2. Move the older report log to a designated location.
3. Run `ssreport` to create a description of the report log (see Section A.6, "ssreport: A Report Log Utility").
4. Archive (and probably compress) the report log and its description when it is no longer needed to generate reports.

You *must* rotate a report log at the following times:

- If you move the license server to a new machine.
If *SAMreport* detects that the license server `hostid` changes in a report log used to generate a report, it stops processing the report log and includes in the report only the data preceding the `hostid` change.
- If a vendor daemon name changes.
If *SAMreport* detects that the vendor daemon name changes in the report log used to generate a report, it stops processing the report log and includes in the report only the data preceding the vendor daemon name change.

You may want to rotate a report log at a regular time interval, such as weekly or monthly, depending on how quickly the report log grows. The rotation of a report log does not need to be tied to the interval over which you generate your reports—remember that you can generate a report on multiple report logs and that you can filter report data with designated start and end dates. You may also want to rotate report logs when the license limit for a feature changes, but alternatively, you can also set start and end date filters to isolate usage for given license limits.

Note: You can corrupt a report log by rotating it incorrectly.

A report log must begin with a header that is written by the vendor daemon. Without this header, the report log is corrupted and the data in the report log is unusable and unsalvagable from the start of the report log until the first time in that report log that the license server is restarted.

The report log will have no header and will be corrupted if you manually copy an active report log to a new file name, delete the original active report log, and keep the vendor daemon writing to the original active report log file name.

A.7.1 Finding Vendor Daemon and FLEXnet Licensing Utilities Versions

How you are able to start and rotate report logs depends on the version of the vendor daemon and `lmswitchr` that you use. The `-switchlog` batch option in `SAMsuite` uses `lmswitchr`.

Find the version of the vendor daemon.

WINDOWS

To find the version of a vendor daemon, start LMTOOLS, the FLEXnet Licensing utility. Select the Utilities tab where you can find the version of FLEXnet Licensing that is built into a file. Enter or browse to the vendor daemon and click the Find Version button. The FLEXnet Licensing version is displayed at the bottom of the LMTOOLS window.

`lmswitchr` is found in LMTOOLS. Check the version of LMTOOLS (and therefore `lmswitchr`) by selecting Help | About.

UNIX

To find the version of your vendor daemon, use:

```
lmutil lmstat path_to_vendor_daemon
```

or

```
vendor_daemon -v
```

`lmswitchr` is part of `lmutil`. All of the `lmutil` FLEXnet Licensing utilities are a single binary whose behavior is determined by its file name. If you do not have `lmswitchr`, but have `lmutil`, copy `lmutil` to `lmswitchr`. Check the version of `lmswitchr` by typing in the directory with `lmutil` and `lmswitchr`:

```
lmutil lmver lmswitchr
```

`lmswitchr` does not work reliably with vendor daemons older than FLEX`lm` v4.1 and does not work at all with v3.0 or v4.0 vendor daemons. If you have a pre-v4.1 vendor daemon, contact your vendor for a newer daemon.

A.7.2 Rotating a Report Log with `lmswitchr`

The `lmswitchr` utility switches the report log file by closing the existing report log and starting a new report log with a new file name. It also starts a new report log file if one does not already exist. This method of rotating a report log is recommended if you are not using `SAMsuite`.

Usage is:

```
lmswitchr [-c license_file_list] feature new_report_log
```

or with v5+ vendor daemon:

```
lmswitchr [-c license_file_list] vendor new_report_log
```

WINDOWS

1. Start `LMTOOLS`. In the Service/License File tab, select Configuration using License File. Click the Switch Report Log tab. Type the Vendor Name and the New Report File Location. Be sure to specify a new file name (do not reuse the old file name) for the new report log. Click the Switch Report Log button. Skip to step 2 below.

UNIX

1. Issue the `lmswitchr` command for a vendor daemon. For redundant license servers, issue an `lmswitchr` command on only one of the machines where the vendor daemon runs (the command acts on the master redundant license server)—issuing `lmswitchr` on each machine where a redundant server runs can cause the loss of the report logs. Be sure to specify a new file name (do not reuse the old file name) for the new report log. If you are using `lmswitchr` v5.1+, type:

```
lmswitchr -c path_to_lic_file vendor_daemon
           path_to_new_rl_name
```

where:

`path_to_lic_file` is the full path to the license file that is currently used by the license server.

If you are using `lmswitchr` v5.0, you may substitute one of the feature names that the vendor daemon manages for `vendor_daemon`.

2. Edit the options file with the new report log path and file name. If the license server goes down and comes back up, the vendor daemon reads its options file and continues to write to the new report log. For redundant license servers, update each of the three options files.
3. Move the closed report log to where it can be accessed for reporting and storage (to the machine where *SAMreport* runs or to a disk that is mounted on the machine where *SAMreport* runs). You move one report log for a single license server and up to three report logs for a redundant license server.
4. Eventually you will not need to generate reports on some of your old report logs. Those report logs can be compressed and archived to a location that does not necessarily need to be accessed by *SAMreport*.

A.7.3 Rotating a Report Log with *lmnewlog*

The *lmnewlog* utility switches the report log file by moving the existing report log information to a new file, then starting a new report log with the original report log file name. If you rotate report logs with *lmnewlog* instead of *lmswitchr*, you do not have to change the file name in the *REPORTLOG* line of the vendor's options file. Requires a v7.1+ vendor daemon.

Usage is:

```
lmnewlog [-c license_file_list] feature  
          renamed_report _log
```

or:

```
lmnewlog [-c license_file_list] vendor  
          renamed_report _log
```

A.7.4 Rotating a Report Log with *SAMsuite*

If you have *SAMsuite*, you can use one of its batch options, *-switchlog*, to rotate a report log.

1. Issue the following batch command to a vendor daemon. For redundant license servers, issue this command only once per vendor daemon. Be sure to specify a file name for the new report log that is different from the report log that is currently getting written.

```
sam -batch -folder folder -switchlog vendor_daemon  
          new_rl_name
```

SAMsuite closes the current report log and starts writing to the new report log. It also changes the report log name stored in the folder in the *SAMsuite* database. *SAMsuite* reinstalls the options file, ensuring that when the license server restarts, the vendor daemon continues to write to the new report log file.

2. Move the closed report log to where it can be accessed for reporting and storage (to the machine where *SAMreport* runs or to a disk that is mounted on the machine where *SAMreport* runs). You move one report log for a single license server and up to three report logs for a redundant license server.
3. Eventually you will not need to generate reports on some of your old report logs. Those report logs can be compressed and archived to a location that does not necessarily need to be accessed by *SAMreport*.

To start a new report log for a v3.0 or v4.0 vendor daemon that cannot use `lmswitchr`, you can change the path to the report log (Setup | Edit | Edit Folder), install the folder, and restart the license server (a reread does not suffice).

A.7.5 Rotating a Report Log without Changing its File Name

If you do not want to change the name of the report log when you rotate it (for example, if you have scripts that expect the report log to have a specific name), we recommend maintaining a symbolic link to the report log and using this link in your scripts. Each time you change the report log name, change the target of the symbolic link.

When using v7.1+ vendor daemons, you can use `lmnewlog` to change the name of the report log instead of the symbolic link option.

A.8 Avoiding Corruption in Report Logs

FLEXnet Licensing has been improved from version to version to make report logging more robust. Nevertheless, particularly in older versions of the vendor daemon, corruption can occur in the report log.

Follow these guidelines to avoid corrupt report log files:

- Run your license servers with the latest available versions of `lmgrd` and your vendor daemons. Newer versions provide bug fixes and enhanced behavior. Regularly visit our web site to check for the latest versions of `lmgrd` and `lmutil`:

<http://www.macrovision.com/services/support/flexlm>

Check with your software vendors for the newest available versions of their vendor daemons.

- In pre-v5.1 vendor daemons, rereads and shutdowns could possibly introduce corruption, particularly if the server is busy. With these older vendor daemons, the best policy is to issue reread or shutdown when no licenses are checked out, if possible.
- Successive `lmdown` commands. Issuing an `lmdown` command while the server is shutting down can cause corruption.
- Don't move, copy, or edit the report log file while the server is running.
- Never configure more than one license server to write to the same report log file. In the case of a redundant server, each vendor daemon must write to a distinct file, preferably one which is local to the server machine, not one which is available through NFS. Even if the report log is not corrupted, *SAMreport* does not generate a report using a report log that has been written by a vendor daemon running on more than one license server machine.
- Never configure different vendor daemons to write to the same report log file. Even if the report log is not corrupted, *SAMreport* does not generate a report using a report log that has been written by more than one vendor daemon.
- Do not use `lmswitchr` or `-switchlog` to switch the report log to an existing report log file. Switch to a file with a new name.
- Never hand-edit a report log or combine report logs together into the same file.

Sample Scripts

Here are two sample scripts, `last_month.csh` and `rotate_log.csh`. These scripts are provided as examples and are not supported by Macrovision.

B.1 `last_month.csh`

The `last_month.csh` script is designed to generate a report for the previous month using all the report logs in a directory.

```
#!/bin/csh
# Date Created: Jan 20, 2000
# This script generates a report for the previous month using
# all the report logs in the directory specified by
# REPORTLOG_DIR.
#
# The output gets placed in a subdirectory containing the month
# and year of the report in the directory specified by OUTPUT_DIR.
#
# The type of report, type of output(s), filters, etc. that are
# the same every time the report is ran go in the file specified
# by CONFIG_FILE.
#
# There are five environment variables you need to either set from
# the command line or modify in the script in the first section
# below:
# This script assumes that SAMreport_home is included
# in $PATH.
# REPORTLOG_DIR - Directory the reports logs used for this report
# are located
# WORKING_DIR - Directory where temporary config files will be
# created
# CONFIG_FILE - Config file containing parameters used every time
# OUTPUT_BASE - Base file name of the output files
# OUTPUT_DIR - Directory which a subdirectory in the form
# month year will be created to contain the generated report
```

last_month.csh

```
# Set the necessary environment variables if they are not already
# set
if (! $?SAMREPORHOME) then
    # SAMreport's directory which contains the "report" script
    setenv SAMREPORHOME="/t/richma/samreport/current"
endif
if (! $?REPORTLOG_DIR) then
    # Directory the reports logs used for this report are located
    setenv REPORTLOG_DIR="/t/richma/last_month/reportlogs"
endif
if (! $?WORKING_DIR) then
    # Directory where temporary config files will be created
    setenv WORKING_DIR=`pwd`
endif
if (! $?CONFIG_FILE) then
    # The config file containing parameters used every time the
    # report is generated
    # File needs to contain parameters for type of report and output
    # type(s).
    setenv \
        CONFIG_FILE="/t/richma/last_month/usage_over_time.config"
endif
if (! $?OUTPUT_BASE) then
    # The base file name of the output files
    setenv OUTPUT_BASE="monthly"
endif
if (! $?OUTPUT_DIR) then
    # Directory which a subdirectory in the form month year will
    # be created to contain the generated report
    setenv OUTPUT_DIR="/t/richma/last_month/web"
endif

# Make sure this umask allows proper file access for viewing
umask 022

# Get the month and year of the previous month
@ month_num = `date +%m` - 1
@ year = `date +%Y`
# If current month is January, need to set month to December
# and decrement the year
if ($month_num == 0) then
    @ month_num = 12
    @ year = $year - 1
endif
```



```

# Now we need to find the number of days in the month for the end
# date
set num_days=(31 28 31 30 31 30 31 31 30 31 30 31)
set end_day=$num_days[$month_num]
# If the month is February...
if ($month_num == 2) then
    @ by_4    = $year % 4          # 0 if divisible by 4
    @ by_100  = $year % 100       # 0 if divisible by 100
    @ by_400  = $year % 400       # 0 if divisible by 400
    # If year is divisible by 4 and not divisible by 100
    # or is divisible by 400 then it is a leap year.
    if (($by_4 == 0 && $by_100 != 0) || ($by_400 == 0)) then
        set end_day=29
    endif
endif

# Put the -startdate and -enddate parameters in the
# startdate.config file
echo "-startdate \"\$month_num/01/$year 00:00:00\" >! \
$WORKING_DIR/startenddate.config"
echo "-enddate \"\$month_num/$end_day/$year 23:59:59\" >> \
$WORKING_DIR/startenddate.config"

# List of months used to create the subdirectory containing the
# output files
set months=(Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec)
# Output the -output parameter and the directory/base to be used
echo "-output
\${OUTPUT_DIR}/${months[$month_num]}$year/${OUTPUT_BASE}\" >! \
$WORKING_DIR/output.config"

# Make the subdirectory in OUTPUT_DIR if it doesn't already exist
# Directory is in the format month year such as Jan2000
if (! -d "${OUTPUT_DIR}/${months[$month_num]}$year") then
    mkdir "${OUTPUT_DIR}/${months[$month_num]}$year"
endif

# Run SAMreport using the config files created above
# NOTE: Be sure to change Summary to Raw if report is of type Raw
report Summary -config "$CONFIG_FILE" \
    -input "$WORKING_DIR/*.rl" \
    -config "$WORKING_DIR/startenddate.config" \
    -config "$WORKING_DIR/output.config" \
    >&! "$WORKING_DIR/sr_errors.log"

```

B.2 rotate_log.csh

The rotate_log.csh script is designed to rotate a report log.

```
#!/bin/csh
#
# Date Created: Feb 01, 2000
#
# This script rotates the report log for the specified vendor
# daemon. It will close out the old report log and start a new
# one by issuing a lmswitchr call. The new report log will be
# in the same directory as the old one, and will be given a file
# name in the form vendor_YYYYMMDD.rl. The file name will also
# be updated in the specified options file.
#
# The following Environment Variables need to either be set on the
# command line before running this script or the first block of
# commands in this script needs to be changed to reflect your
# license server's environment:
# LMUTIL - The path and file name of lmutil
# LICENSE_FILE - The path and file name of your license file
# VENDOR_DAEMON - The name of the vendor daemon (e.g. gsi)
# OPTIONS_FILE - The path and file name to this vendor daemon's
# options file

# Set the necessary environment variables if they are not already
# set
# (Change these to fit the environment of the license server which
# has a report log you wish to rotate, or set on the command line)
if (! $?LMUTIL) then
    # The path and file name of lmutil
    setenv LMUTIL `which lmutil`
endif
if (! $?LICENSE_FILE) then
    # The path and file name of your license file
    setenv LICENSE_FILE "/usr/local/flexlm/license.dat"
endif
if (! $?VENDOR_DAEMON) then
    # The name of the vendor daemon (e.g. gsi)
    setenv VENDOR_DAEMON "vendord"
endif
if (! $?OPTIONS_FILE) then
    # The path and file name to this vendor daemon's options file
    setenv OPTIONS_FILE "/usr/local/flexlm/vendord.opt"
endif
```

```

# Get the path to the old report log
setenv OLD_REPORT_LOG \
`awk '/^REPORTLOG/{print $2}' $OPTIONS_FILE`
# Remove the + from the path/filename
setenv OLD_REPORT_LOG `echo $OLD_REPORT_LOG | sed 's/^\+//`
# Need to remove the .rl extension to create the new report log
# path/filename
setenv NEW_REPORT_LOG `echo $OLD_REPORT_LOG | sed 's/\.rl$//`
# Need to remove the _YYYYMMDD part of the old file name
setenv NEW_REPORT_LOG \
`echo $NEW_REPORT_LOG | sed 's/_[0-9]\{8\}$//`
# Create the new report log file name with today's date
setenv NEW_REPORT_LOG $NEW_REPORT_LOG"_"`date +%Y%m%d`.rl"
# Make sure that the report log hasn't already been rotated today
if($OLD_REPORT_LOG == $NEW_REPORT_LOG) then
    echo "The report log has already been rotated today."
    exit 1
endif

# Run "lmutil lmswitchr" to switch the report log the vendor
# daemon is writing
$LMUTIL lmswitchr -c $LICENSE_FILE $VENDOR_DAEMON
NEW_REPORT_LOG

# We need to quote forward slashes with backslashes for the next
# sed command
setenv NEW_REPORT_LOG `echo $NEW_REPORT_LOG | sed 's/\/\\\/\\\/g`
# Backup the old options file
mv $OPTIONS_FILE $OPTIONS_FILE.old

# Create a new options file with the old, replacing the REPORTLOG
# line
sed "s/^REPORTLOG .*/REPORTLOG +$NEW_REPORT_LOG/" \
$OPTIONS_FILE.old > $OPTIONS_FILE

echo "Done. Report log being written to is now $NEW_REPORT_LOG. \
Options file has been updated."

```

rotate_log.csh

Launcher and Java Options

In addition to the command-line options to the `report` command that control the type of report to be generated, options that control Java Virtual Machine behavior are available.

C.1 Options for the SAMreport Launcher on Windows

The Java VM-specific options to the SAMreport launcher on Windows are listed in Table C-1. They follow, and are separated from, the `report` options by a double dash, `--`. Any options that follow the `--` are passed directly to the Java VM and are not recognized by SAMreport.

```
report [-- launcher_args]
```

Table C-1: Java VM-Specific Options to the SAMreport Launcher on Windows.

Launcher Argument	Description
<code>-c[onsole]</code>	Run in console mode.
<code>-gsipackage name</code>	Set the package prefix.
<code>-javaopts options</code>	Set command-line options for the Java VM.
<code>-javaprogram prog_name</code>	Sets the Java program.
<code>-javaver version</code>	Specify which version of Java to use.
<code>-l log_file</code>	Create a launch log file.
<code>-usecwd</code>	Use current directory as SAMreport home.

Table C-1: Java VM-Specific Options to the SAMreport Launcher on Windows.

Launcher Argument	Description
-w[ait]	This option instructs the launcher not to exit until report execution has finished. Helpful for running from scripts.

To display the current launcher options, type:

```
report -h
```

C.2 Specifying Options to the Java VM

WINDOWS

As you saw in Table C-1, options to the Java VM on Windows can be specified to the SAMreport launcher using the following syntax:

```
report [report_family report_args]
      -- -javaopts [java_options]
```

UNIX

Options can be specified to the Java VM on UNIX by setting the JAVA_OPTIONS environment variable for the user who runs the report command:

```
% setenv JAVA_OPTIONS [java_options]
```

C.3 Specifying Initial and Maximum Java Heap Size

The Java heap is the RAM reserved for use by the Java Virtual Machine. At any given time, only a portion of the heap is in use and the size of that portion varies over time. If the size of the used portion approaches the size of the heap, then more memory is allocated to the heap. If the size of the used portion falls, the size of the heap may be reduced by the VM, to free memory for the operating system. The maximum Java heap size is the maximum amount of memory available to SAMreport. If more memory than the maximum heap size is needed by SAMreport, then an OutOfMemoryError is displayed in the Status History pane, SAMreport_home\samlog.txt (Windows), or a terminal window (UNIX).

The default initial Java heap size is set to 64MB, except on SGI where it is set to 20MB. The default maximum Java heap size is set to 100MB on all platforms. The maximum heap size should be no more than 80% of the total RAM of the machine.

If you want Java heap sizes other than the defaults, you must specify them when you start *SAMreport*. This can be done as a command-line option to the Windows launcher or in `JAVA_OPTIONS` settings on UNIX. The initial heap size is set with `-Xms` and the maximum heap size is set with `-Xmx`. For example, to increase the maximum Java heap size to 200 MB:

WINDOWS

Set Java options on the *SAMreport* launcher:

```
report [report_family report_args]
      -- -javaopts -Xmx200m
```

UNIX

Set `JAVA_OPTIONS` for the user who runs the `report` command:

```
% setenv JAVA_OPTIONS -Xmx200m
```

Then run the `report` command:

```
% report [report_family report_args] &
```

C.4 Switching Java Versions

If you have a default version of the JRE that runs on the *SAMreport* machine different from the version of the JRE recommended for *SAMreport*, you can specify a version of the JRE for *SAMreport* to use.

WINDOWS

Install the recommended version of the JRE for *SAMreport*. To override a different default version of the JRE, do one of the following:

- Make a new shortcut for running the *SAMreport* interface using the recommended version with the following command.

```
"C:\Program Files\Macrovision\samreport\version\report.exe"
  -- -javaver 1.4.1_01
```

- Type the command to start a command-line report. For example, to use Java 1.4.1_01 for *SAMreport*, if you have a newer version of the JRE installed:

```
report -- -javaver 1.4.1_01
```

UNIX

When you installed *SAMreport*, you specified the directory for the version of Java you use most often (this directory was saved in the *SAMreport_home*/machind/bin/whereisjava file). If you always want to run with a different version of the JRE, edit the whereisjava file to point to the correct JRE version.

To temporarily run with a different JRE, set the JAVADIR environment variable to the directory containing the alternative Java version. For example:

```
% setenv JAVADIR /usr/local/j2dk1.4.1_01
% report &
```

When you are done using the alternative Java version, unset the JAVADIR environment variable to revert to the Java version specified in the whereisjava file:

```
% unsetenv JAVADIR
```


Command-Line Report Reference

Guidelines for generating reports from the command line can be found in Chapter 8, “Generating Command-Line Reports.”

D.1 Command-Line Arguments

Table D-1 shows the command-line arguments available for the `report` command. Arguments that must be set are marked as *Mandatory*.

Table D-1: SAMreport Command-Line Arguments

Command-Line Argument	Description
<code>{Raw Summary Server}</code> <code>-help</code>	Command-line help for the <code>report</code> command that lists command-line argument usage and the valid values of the <code>-breakdown</code> and <code>-filter</code> arguments.
<code>-alias type from to</code>	Display the specified feature or user name alias in report output instead of the feature or user name recorded in a report log. <i>type</i> is feature or user; <i>from</i> is the feature or user name recorded in a report log; <i>to</i> is the alias to be displayed in report output for the feature or user name. It is possible to map more than one user name to the same alias. If you have v5+ report logs, it is also possible to map more than feature name to the same alias.

Table D-1: SAMreport Command-Line Arguments (cont.)

Command-Line Argument	Description
<code>-barchart <i>metric</i></code>	Specify barchart output for Summary Barchart and Usage Efficiency reports. Available metric values are <code>num_used</code> , <code>%_used</code> , and <code>hrs_used</code> , <code>max_used</code> .
<code>-bgcolor <i>color</i></code>	Set the background color of a graphical report. Select from: white, black, red, green, blue, cyan, magenta, yellow. Default: white.
<code>-breakdown <i>category</i></code>	Specify a breakdown category. Breakdown categories are listed in a later table.
<code>-c <i>license_file_path</i></code>	Specify a license file for SAMreport different from the one located at <code>SAMreport_home/machind/license/license.dat</code> .
<code>-config <i>config_filename</i></code>	Read report parameters from a report configuration file.
<code>-coverage</code>	Calculate Server Coverage data. Syntax is: <code>report Server -coverage</code>
<code>-datacolor <i>color[,color]*</i></code>	Set the color(s) of lines or bars of data 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.

Table D-1: SAMreport Command-Line Arguments (cont.)

Command-Line Argument	Description
<code>-datastyle n[,n]*</code>	Set the style of line(s) of data. Select from -1 (solid), 0 (dashed), 1 (dotted), or 2 (dotted-dashed). Default: -1.
<code>-delim "delimiter"</code>	Specify a RIF format field delimiter. <i>delimiter</i> can be <code>tab</code> , <code>:</code> , or an arbitrary <i>string</i> . The default delimiter is a space. Any fields that contain the delimiter are quoted with double quotes. Do not use double quotes (<code>"</code>), single quotes (<code>'</code>), or a backslash (<code>\</code>) as a RIF delimiter.
<code>-elim_false_denials [interval]</code>	Request the elimination of “false” denials occurring within an interval of given number of seconds. If not explicitly specified, <i>interval</i> is 10 seconds. See Section 6.5, “Eliminating “False” Denials.”
<code>-encoding charset</code>	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.
<code>-enddate date</code>	Set the ending date for report output.
<code>-event_type type</code>	Specify for each event type to include in report. Available event types are <code>used</code> , <code>denied</code> , <code>queued</code> , <code>exception</code> , and <code>checkout</code> . If <i>event_type</i> is not specified, all event types except <code>checkout</code> are reported.
<code>-feat_uniq id</code>	Distinguish features using one of these <i>id</i> 's. Available values for <i>id</i> are <code>name</code> , <code>name_ver</code> , <code>name_key</code> , <code>name_asset</code> .

Table D-1: SAMreport Command-Line Arguments (cont.)

Command-Line Argument	Description
<code>-field field_name [,col_width] [,col_justification]</code>	Include a field with these specifications in a Raw report. Field names are listed in a later table.
<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 <code>d.hh:mm:ss</code> , where <code>d</code> is 1-7 (Day 1 = Sunday, Day 7 = Saturday). Designating days and seconds is optional. Surround <code>start</code> and <code>end</code> 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 font</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. The default value is 900 seconds (15 minutes).
<code>-h</code>	Command-line help for Windows launcher.

Table D-1: SAMreport Command-Line Arguments (cont.)

Command-Line Argument	Description
-html	Request report output in HTML format.
-hwmot [na]	Calculate High Water Mark data. Use the na flag to omit displaying in the report the line that shows available licenses.
-input <input_file_name1 </input_file_name1 [<input_file_name2...]< td=""> <td>Specify the list of report log(s) to read. The -input argument can be used once followed by several report logs. <i>Mandatory</i> (or -input_single)</td> </input_file_name2...]<>	Specify the list of report log(s) to read. The -input argument can be used once followed by several report logs. <i>Mandatory</i> (or -input_single)
-input_single "file name"	Specify a single report log for input. Useful for specifying a file name containing white space on Windows. Surround file name with double quotes.
-interval <i>period</i>	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.
-jpeg [<i>width height</i>]	Request graphical report output in JPEG/HTML format. Optionally, set the dimensions of the JPEG images in pixels. Default: 600 x 600.
-linewidth <i>width</i>	Set the thickness of lines in the graph in pixels. Affects both data and axes. Default: 1.
-no_auto_reduce	Used to circumvent automatic data reduction on Usage Over Time and High Water Mark graphs. Not recommended.
-orderby <i>items</i>	Sort categories are f (feature) and b (breakdown). More than one sort category can be specified at a time. For example, -orderby fb sorts first by feature, then by breakdown category.

Table D-1: SAMreport Command-Line Arguments (cont.)

Command-Line Argument	Description
-output <i>output_filename_base</i>	Specify the output file name base. Do not include a file name extension. <i>Mandatory</i> for Text, HTML, RIF, JPEG/HTML, and PNG/HTML output.
-periodic	Calculate Summary Barchart or Usage Summary for time periods. Use in conjunction with <i>-interval period</i> .
-png [<i>width height</i>]	Request graphical report output in PNG/HTML format. Optionally, set the dimensions of the PNG images in pixels. Default: 600 x 600.
-progress	Shows progress of report generator when running from the command line. On Windows, writes progress as percentage completion to <i>samlog.txt</i> file.
-rif	Request report output in Report Interchange Format (RIF).
-rlwarnings <i>level</i>	Suppress report log warnings associated with vendor daemon to the given <i>level</i> . 0: Show all warnings 1: Suppress abrupt session termination warnings
-showzero	Include entries for features with no usage. Available for Usage Over Time, High Water Mark, Summary Barchart, and Usage Summary reports.
-startdate <i>date</i>	Set the beginning date for report output.
-text	Request report output in plain text format.

Table D-1: *SAMreport* Command-Line Arguments (cont.)

Command-Line Argument	Description
<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.
<code>-timeval_seconds</code>	Show time value in RIF files as UNIX seconds since 00:00:00 January 1, 1970 GMT.
<code>-title title</code>	Set the title on a graphical report.
<code>-uot [na] [pct]</code>	Calculate Usage Over Time or High Water Mark data. Use the <code>na</code> flag to omit displaying in the report the line that shows available licenses. Use the <code>pct</code> flag to display usage in percentages of available licenses instead of count.
<code>-usage_efficiency</code>	Calculate Usage Efficiency data—the incremental usage of licenses.
<code>-usage_summary</code>	Calculate Usage Summary data.

D.2 Breakdown Categories

Table D-2 shows the breakdown categories available for *SAMreport* from the command line.

Table D-2: *Breakdown Categories Available from the Command Line*

Breakdown Category	Description
<code>exp_date</code>	Expiration date on FEATURE/INCREMENT line (string with wildcards).
<code>feat_hostid</code>	HOSTID on FEATURE/INCREMENT line.

Table D-2: Breakdown Categories Available from the Command Line (cont.)

Breakdown Category	Description
group	GROUP name in end-users options file for vendor. If you break down usage by group, you should either: <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 Licensing from v10.0 on) in lmgrd, vendor daemon, and application. For example, sun4_u5 (Solaris), i86_n3 (Win95/NT).
user_product_version	Version of product 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.
version	Version on FEATURE/INCREMENT line.

D.3 Filter Categories

Table D-3 shows the filter categories supported by *SAMreport*, as long as your report log contains data in the category you select. In general, the later the FLEXnet Licensing version of the vendor daemon that wrote the report log, the more of these categories are available in the report log to filter on.

Table D-3: Filter Categories Available from the Command Line

Filter Category	Description
asset_info	asset_info= <i>text</i> on FEATURE/INCREMENT line.
dist_info	dist_info= <i>text</i> on FEATURE/INCREMENT line.
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= <i>dd-mmm-yyyy</i> on FEATURE/INCREMENT line.
issuer	ISSUER= <i>text</i> on FEATURE/INCREMENT line.
license_key	License key on FEATURE/INCREMENT line.

Filter Categories

Table D-3: Filter Categories Available from the Command Line (cont.)

Filter Category	Description
notice	NOTICE= <i>text</i> 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= <i>nnn</i> 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= <i>text</i> on FEATURE/INCREMENT line.
user_ipaddr	User IP address (<i>nnn.nnn.nnn.nnn</i>).
user_platform	Platform the application was compiled to run on. Requires FLEXlm v5+ (called FLEXnet Licensing from v10.0 on) in lmgrd, vendor daemon, and application. For example, sun4_u5 (Solaris), i86_n3 (Win95/NT).
user_product_version	Version of product actually used.
user_project	User LM_PROJECT environment variable (or Windows registry).

Table D-3: Filter Categories Available from the Command Line (cont.)

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= <i>text</i> or " <i>text</i> " on FEATURE/INCREMENT line.
version	Version on FEATURE/INCREMENT line.

D.4 Raw Report Fields

Table D-4 shows the fields available in a Raw report from the command line or in a configuration file. These fields cannot be selected in the SAMreport GUI. A subset of these fields is defined in `SAMreport_home\machind\samples\raw.config`, the configuration file used by default when a Raw report is generated from the SAMreport GUI.

Table D-4: Raw Report Fields Available at the Command Line

Field Name	Description
count	Number of licenses in request.
duration	Amount of time event lasted.
event	Type of event.
event_end_time	End time of event.
event_time	Start time of event.
group	GROUP name in end-user options file for vendor.

Table D-4: Raw Report Fields Available at the Command Line (cont.)

Field Name	Description
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 report (see Section 3.7.1, “Displaying Duplicate Grouping Information in Raw Reports”).
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.

Table D-4: Raw Report Fields Available at the Command Line (cont.)

Field Name	Description
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 Licensing from v10.0 on) in lmgrd, vendor daemon, and application.
user_product_version	Version of product actually 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.

Raw Report Fields

SAMreport Error Messages

The following runtime error messages may be written to:

- The Status History pane of the SAMreport interface
- The Report Log Processing Errors section of a Server Coverage report
- A terminal window (UNIX) or *SAMreport_home*\samlog.txt (Windows) during command line processing

```
Error: Report log has been corrupted (-4) file line no  
(id 50)(minor -1) (Invalid key found, record lost)
```

Corrupt report log event found in report log. May be associated with a vendor daemon crash.

```
Error: Internal Error (-7) file line no  
(id 54)(minor 0) feature key
```

Indicates an unexplained problem in SAMreport. This error should be reported to Technical Support.

```
Error: A reportlog is missing (-9) file line no  
(id 40)(minor 0) feature key
```

Report log contains a gap of significant duration between an end (or lmswitchr) and the following start event.

```
Error: Reportlog is truncated (-11) file line no  
(id 48)(minor -5) (Vendor daemon crashed and restart)
```

Report log event fragment is seen immediately followed by a restart. The feature corresponding to the event fragment is undetermined. Usually caused by vendor daemon crash.

Error: Reportlog is truncated (-11) *file line no*
(id 48)(minor -5) (Vendor daemon crashed and restart) *feature*
key

Report log event fragment is seen immediately followed by a restart.

The feature corresponding to the event fragment is shown. This event is unusable. Usually caused by vendor daemon crash.

Error: Reportlog is truncated (-11) *file line no*
(id 42)(minor 0) *feature key*

Report log file ends abruptly. Probably caused by reading or copying the report log while the vendor daemon is running. Could also be the result of a vendor daemon crash with no subsequent restart.

Error: An error was found in the reportlog (-13) *file line no*
(id 51)(minor -4) (Checkout occurred without checkin, recovery attempted) *feature key*

No matching checkin event found for a checkout event. Because the error is reported much later than the actual error occurred, the line number is not particularly relevant. Sometimes this error is reported in error.

Error: An error was found in the reportlog (-13) *file line no*
(id 37)(minor -7) (A data field is corrupted, record recovered, field recovery attempted) *feature key*

Historically seen only in buggy, pre-v5 report logs. Indicates corruption in one of the report event fields.

Error: Countby group requested, but event belongs to more than one group (-17) *file line no*
(id 61)(minor 0) *user:(group1, group2)*

To avoid overreporting usage, if a user is a member of more than one group, only one of the groups is arbitrarily chosen for the purposes of presenting the information about a checkout by that user, and an error is issued.

To avoid this error, filter out users who are in more than one group, or filter out all but one of the groups who have users in common.

Error: End date may be using 2 digit year instead of 4 digits
Error occurred: com.macrovision.samreport.JrepException:
Unable to generate report.

This is an example of an error that occurs when a parameter or an argument is invalid and a report cannot be generated. The first line describes which parameter or argument is invalid.

Error: In *file* vendor changed from *vendor_name1(hostid1)* to *vendor_name2(hostid2)*. No further events will be read from this file.

This report log has been written by more than one vendor daemon or by a vendor daemon running on more than one *hostid*.

Error reporting threshold of 2000 has been reached. No further errors will be reported.

Error: Inconsistencies found in report log around *file* line *no*

Inconsistency in a block of data within a report log has been detected, for example when data is missing, altered, or corrupt.

Error: Unable to load file *file*. File is too large.

This error occurs when the report output is too large to load into the SAMreport Viewer (larger than about 500kb). There are several factors that contribute to the size of report output, for example, large report log, especially with a Raw report, small summary period, etc.

Corruption: Timestamp backups were seen, possibly due to report log corruption.

This error indicates that timestamp backups larger than the threshold value (three minutes) were seen in the report log. Timestamp backups under the threshold are noted with a warning.

Corruption: Significant corruption in the input report log(s) has rendered parts of this report suspect.

This error indicates that a negative duration or negative number of licenses has caused significant errors in the report.

Warning: Unable to read report log *file*: Report log version is newer than report generator. Please upgrade report generator before reading this file.

This warning indicates the major version of the report log is newer than the report generator.

Warning: Reportlog session terminated abruptly (-11) *file* line *no* (id 42)(minor 0) *feature key*
java.lang.OutOfMemoryError

This warning indicates that the Java Virtual Machine has run out of memory and that the report cannot be generated. See instructions in Section C.3, “Specifying Initial and Maximum Java Heap Size,” to increase the memory available to Java.

Non-fatal error occurred while reading *file*: Warning: Report log version (*n.x*) is newer than the report generator (*n.y*). Please upgrade your report generator.

This warning indicates that the minor version of the report log is newer than the report generator. A report is generated.

Index

A

- about SAMreport
 - Java version 15
 - SAMreport version 15
- Advanced pane 18
- arguments, command-line 113
- Arrange pane 17

B

- breakdown categories at command line 119
- breakdown mask 78
- building configuration files 71

C

- checkout event 42
- combining report logs 47
- command line
 - arguments 113
 - breakdown categories 119
 - filter categories 121
 - Raw Text report fields 123
- configuration files
 - building 71
 - list of commands 74
 - loading 71
 - saving 71

D

- date formats, for reports 70
- denials in Raw reports 44

- denials, false 68
- denials, ultimate 46, 69
- denied event 68
- duplicate grouping 42, 59

E

- eliminating false denials 68
- enabling a report log 91
- error messages 127
- event type
 - checkout 42
 - denied 68
 - exception 68
 - queued 67
 - unsupported 68
 - used 67
- event type filter 17
- exception event 68

F

- false denials, eliminating 68
- features, distinguishing
 - name and asset info 58
 - name and license key 58
 - name and version 58
- fields available in Raw Text reports 123
- filter categories at command line 121
- Filter pane 17
- flush data to report log 95
- formats, date 70

G

gap 38
generating report 11
graphs
 see reports, High Water Mark
 see reports, Usage Over Time
group, in Raw Text reports 41

H

help, command line
 for report command 74, 113
 for Windows launcher 116
High Water Mark report
 see reports, High Water Mark
hostid, identify in report log 15

J

Java
 specifying heap sizes 110
 specifying options 110
 specifying version 111

L

launcher on Windows 109
license file, options file path 92, 93
license limit, usage above 33
license server, identify in report log 15
license sharing 59
list of commands 74
lmnewlog 100
lmreread 95
lmswitchr 94, 99
loading configuration files 71

M

manage report log 96
merging report logs 47

O

options file, enable a report log in 91, 93

overdraft license usage 32

P

pane
 Advanced 18
 Arrange 17
 Filter 17
 Status History 19
preface vii

Q

queued event 67

R

Raw Text report
 see reports, Raw Text
reason for denials 87
redundant servers, report logs 47, 91
report log 91
report log file
 from redundant servers 47, 91
 generating report 11
 manage 96
 size 94
 time zones 66
 using multiple 47
reports
 combining report logs 47
 date formats 70
 errors/warnings 23
 event type filter 17
 High Water Mark 27
 overdraft license usage 32
 Raw Text 40
 fields 123
 group 41
 reason for denials 44, 87
 why denials 44, 87
 Server Coverage 38
 show zero usage 17

- start and end dates 70
- Summary Barchart 29
- usage above license limit 33
- Usage Efficiency 31
- Usage Over Time 25
- Usage Summary 33
- using multiple report logs 47
- reread license file 95
- rotate report log 96

S

- samlog.txt 74
- SAMreport
 - starting 13
 - stopping 14
- SAMreport launcher 109
- SAMreport launcher on Windows 109
- SAMreport_home directory 13
- saving configuration files 71
- Server Coverage report
 - see reports, Server Coverage
- setting a breakdown mask 78
- show licensing 15
- show zero usage 17
- size of report log 94
- specifying
 - Java heap sizes 110
 - Java options 110
 - Java version 111
- ssreport 95
- starting SAMreport 13
- Status History pane 19
- stopping SAMreport 14
- Summary Barchart report
 - see reports, Summary Barchart

T

- technical support x
- time zones 66
- turning on a report log 91

U

- ultimate denials 46, 69
- unsupported event 68
- Usage Efficiency report
 - see reports, Usage Efficiency
- Usage Over Time report
 - see reports, Usage Over Time
- Usage Summary report
 - see reports, Usage Summary
- used event 67

V

- version
 - Java 15
 - SAMreport 15

W

- why denials 44
- Windows launcher 109

