Sentinel® RMS

Release Notes

v8.4.1 for HPUX Itanium

SafeNet®

© Copyright 2011, SafeNet, Inc. All rights reserved.
http://www.safenet-inc.com

We have attempted to make these documents complete, accurate, and useful, but we cannot guarantee them to be perfect. When we discover errors or omissions, or they are brought to our attention, we endeavor to correct them in succeeding releases of the product.

SafeNet® and Sentinel® are registered trademarks of SafeNet, Inc. All other product names referenced herein are trademarks or registered trademarks of their respective manufacturers.

Part Number 007-009754-001, Revision C
October 2011

Contents

About This Document 5

Product Overview 5

What's New in This Release? 5

Alternate Workflow of Network License Revocation 5

New Architecture for Commuter License Persistence Data 7

Few Guidelines for Migrating Your Existing Commuter Persistence Data 7

Increased Size of Private Vendor Information 7

Longer Private Vendor Information in Commuter Licenses 8

Longer Private Vendor Information in Grace Licenses 8

Grace License Related Enhancements 8

Preventing Grace License Usage When Network License is Obtained 8

Ignoring Grace License in Case of No-net Setting (WT# 83498) 8

Introduced Version 13 Licenses 8

Makefile for Customization of RMS Components 8

Remote Checkout Days Now Extensible (WT#84048) 9

Configuration File to Enable/Disable Fingerprint Caching (WT#84181) 9

Online Documentation 9

Problems Corrected in This Release 10

Known Problems 12

Compatibility and Upgrade Information 13

Changes in the Separator used for IP Addresses and Host Names in LSHOST 13

Compatibility Information for Version 13 Licenses 13

Installation Information 14

Before You Install... 14
System Requirements 15
Installation Instructions 15
Installed Directories 16
Notes: 16
License Revocation FAQs 17
Technical Support 18

About This Document

This document contains details about the features and installation of the Sentinel RMS SDK v 8.4.1 release on HPUX Itanium. This is a service pack release and replaces any previous release.

Product Overview

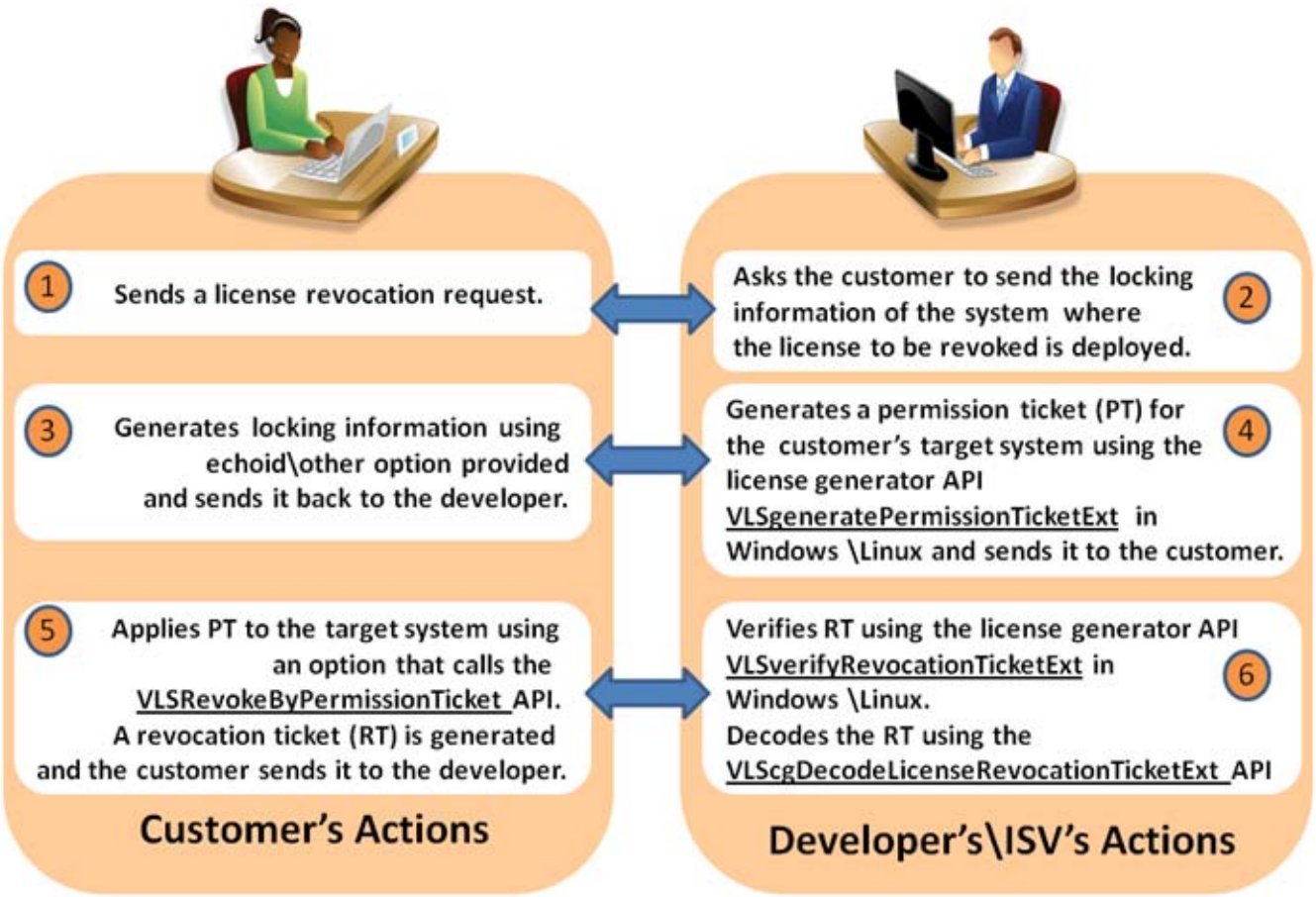
Sentinel RMS is a software licensing SDK for your applications. It increases revenue by authorizing the use of your applications and offers a variety of licensing schemes to boost your product sales. It also provides tools that system administrators can use to track and manage licenses in a network.

What's New in This Release?

This section provides information about the new features and problems corrected in this release.

Alternate Workflow of Network License Revocation

An alternate workflow, shown in the diagram below, is provided for revoking RMS network licenses. This workflow now matches the stand-alone license revocation process.



Below are the related highlights and enhancements:

- You can use the VLSgeneratePermissionTicketExt and VLSverifyRevocationTicketExt APIs on Windows or Linux[1] to generate permission tickets and verify the returned revocation ticket, respectively.
- The VLScgDecodeLicenseRevocationTicketExt API is provided in the lsdecod library of HPUX Itanium to decode revocation tickets.
- The VLSRevokeByPermissionTicket API is updated to cater for network revocation requirements and there is no change in the API signature.
- Backward compatibility is maintained for older implementations.
- In addition to normal licenses, the following can also now be revoked:
  - Capacity-enabled network licenses
  - Trial licenses (stand-alone and network)
- The permission ticket can now contain custom vendor defined data as well. This new field will be helpful for additional verification of the revocation operation.
- The licenses can be revoked regardless of whether they exist in the system or not (that is, even prior to the installation on the license server/system).

1. The license generation module is not available in the Sentinel RMS SDK for HPUX Itanium.

For more information, please refer to the Chapter “License Revocation” of the *Sentinel RMS SDK Developer’s Guide*.

New Architecture for Commuter License Persistence Data

Enhancements have been made in maintaining the license server-side persistence data for commuter licenses. The layered architecture (used so far for storing and managing trial, volume, and stand-alone revocation persistence data) is now also used for the commuter persistence data. As a result, the commuter licenses are now more reliable and provide better performance in handling large data.

Few Guidelines for Migrating Your Existing Commuter Persistence Data

- Upgrading the license server to v8.4.1 or later will migrate the persistence data automatically.
- Persistence data from v8.2.1 to 8.2.2 will be migrated.
- Migration of persistence data is a one-time activity. The license server may take some time to complete this activity—depending upon the size of the database and availability of the system resources.

Note: If you are facing issues while migrating large commuter persistence data, contact Technical Support for assistance.

- During the migration process, the license server will not serve any requests. Therefore, it is recommended to keep the license server unavailable for licensing requests\updates activities during this period.
- Once initialized, the commuter persistence data will be read from/written to the new database only.

Increased Size of Private Vendor Information

The private vendor information can now be up to 2000 bytes long. Earlier, this limit was 395 bytes, inclusive of the public vendor information. A summary of the new limits for various licenses is given below:

Vendor Information - Maximum Size in Various License Codes			
Private	**Long**[a]	**Short-numeric**	**Short**[b]
	2000	98	Not allowed
Public	395	Not allowed	Not allowed

a. In the case of redundant licenses, the vendor information, both public and private, should each not exceed 395 characters.

b. Vendor information cannot be included in short license codes, which were available only prior to v 8.1 (version 11 licenses).

Note: With increased vendor information support, the license code (long readable) can be up to 5200 bytes in length. Earlier, this limit was up to 1024 bytes. The corresponding macro (VLS_MAX_LICENSE_SIZE) in *lserv.h* is updated accordingly.

Longer Private Vendor Information in Commuter Licenses

The commuter licenses can now contain 883 bytes of private vendor information. Earlier, this was limited to 98 characters.

Longer Private Vendor Information in Grace Licenses

Prior to this release, the grace code installation failed on the customer's computer if you have specified lengthy values for properties like feature name, feature version, vendor information and secret text. Only private vendor-information of approximately 64 bytes was supported in grace code. However, now the maximum length/bytes of vendor-information that can be included in grace code is 465 (remaining bytes will be truncated).

Grace License Related Enhancements

The following enhancements are for the grace licenses:

Preventing Grace License Usage When Network License is Obtained

A grace license is meant to provide continued application access in cases like network down-time. Currently, when a grace license is in use, no check is made for license token availability on network till the grace period lasts. Now, you can alter this behavior so that a periodic check is made for license token availability.

This can be done by calling the LSUpdate API. Its second parameter, which was unused so far, is replaced by *ulGraceSwitchToNetworkTm*. Using this new parameter, you can configure a time interval after which the LSUpdate call will look for a license token on network. If a license token is obtained, the application switches from the grace to network license. If the license is not obtained[2], the application continues to run till the grace period lasts. Refer to the *Sentinel RMS SDK API Reference Guide* for more details about using the API.

If you have used auto-update mechanism in your application to automatically renew licenses, then the network license check will be made after every 10 minutes.

Ignoring Grace License in Case of No-net Setting (WT# 83498)

A network license can have grace period associated with it. However, when such a grace-enabled network license is forcibly limited to a stand-alone system[3], you may want to prevent automatic switching to a grace license. This can be done using the new API VLSsetGraceRequestFlag (see the *Sentinel RMS SDK API Reference Guide*). This is useful when you want to ignore grace licenses to run application in stand-alone licensing environment.

Introduced Version 13 Licenses

Due to the increased length of the private vendor information and other changes in RMS, version 13 licenses are introduced. See also, *"Compatibility and Upgrade Information" on page 13*.

Makefile for Customization of RMS Components

A new Makefile is provided to allow easier customization of license server, custom locking, and other features (as described in "Appendix B - Customization Features" of the *Sentinel RMS SDK*

2. The license may not be obtained under various scenarios, even when network connectivity is restored. Such as when license tokens are finished or system locking mismatches, and so on.

3. By setting no-net either in the VLSsetContactServer API or via LSHOST or LSFORCEHOST environment variables.

API Reference Guide). The Makefile can be found in the *examples* directory of your SDK installation. In addition, two other files—*custlock.c* and *custexlock.c*—are provided to facilitate customization of standard custom and extended custom locking.

Remote Checkout Days Now Extensible (WT#84048)

The duration of a remote checked out license can be extended before its expiration. There are two ways to do this:

- **Using the VLSgetCommuterCode API**
  The VLSgetCommuterCode API is now enhanced to allow extension of duration of a remotely checked out license. You can provide an option to your application user via some utility or an option that embeds the VLSgetCommuterCode API.
- **Using lcommute**
  lcommute provides -e option to extend the duration of a remote checked out license.

Note: If values are specified for both the remote commuter extension days and commuter days, the former supersedes the latter.

Refer to the *Sentinel RMS SDK System Administrator's Help* for details on using lcommute.

Configuration File to Enable/Disable Fingerprint Caching (WT#84181)

The default fingerprint caching created problems in case of standard and custom extended locking. To overcome this, a configuration file can be used to set the fingerprint caching as ON or OFF.

Refer to "Appendix B - Customization Features" of the *Sentinel RMS SDK API Reference Guide* for more details.

Online Documentation

The Sentinel RMS Developer's and API Reference Guides are available in cross-platform Web browser based online format. You can launch the guides using the Documentation Access Page or from the *Manuals* folder.

Problems Corrected in This Release

In this release, the following problems are corrected:

WT/Task Ref#	Problem Description
85060	The lserv process terminated while obtaining the disk ID based locking information. The problem is corrected in this release.
84918	The following APIs were missing from the lsdecode library. These are added since the 8.4.1 release: ❑ VLScgDecodeLicense - Decodes the license string and puts the corresponding codeT struct in the last argument ❑ VLScgInitialize - Initializes and allocates resources for generating licenses. This function must be called before using any other VLScgXXX function.
84018	The RMS license server can now start up in absence of network on HPUX Itanium. It now communicates on a local host and honors all the stand-alone licensing features. Earlier, on startup, the RMS license server used */etc/hosts*. Now, this dependency is removed. ***Note:*** The following problem was originally addressed in the 8.2.2 release. However, it had recurred thereafter.
84936	A shared license token was not granted to the application in a scenario when no tokens were left and the contact server was not set (using LSFORCEHOST or VLSsetContactServer). The problem is corrected in this release.
83950	The Sentinel RMS server showed startup messages on standard error (STDERR) instead of standard output (STDOUT).
84994	The memory overruns of the LSUpdate API are corrected in this release.
84464	The *lsapiw32.lib* and *lsdecod32.lib* libraries can now be successfully linked in the same application.
84867	The VLSbatchUpdate API no longer fails due to time conversion issues handled by the library internally.
84748	On a 64-bit system, the VLSgetFeatureInfo API returned incorrect birth\death date for a license with no expiry. The problem is corrected in this release. Now, -1 is returned for such licenses (-1 denotes that no birth\death date is set.).
84857	The license file will remain accessible to the remaining non-admin users if any particular user deletes a license using an RMS API. Earlier, access to the file was limited to the user who deleted a license from the file.
84878	The group reservation file failed to accept more than one subnets specified. The problem is corrected in this release.
84051	The VLSbatchUpdate API—when called to update handles for expired or exhausted grace licenses—returned LS_SUCCESS for each handle. The problem is corrected in this release.
84154	The license update call succeeded even when the IP address was changed at run-time. The problem is corrected in this release.
83994	In a previous release, the signature of the function VLSgetLicenseInfo was modified. However, the macros VLSgetLicenseInfoByIndex and VLSgetLicenseInfoByName that use this API were not updated. Hence, VLS_CALLING_ERROR was returned. Now, correct signatures are included in the *lserv.h*.

WT/Task Ref#	Problem Description
83910	The licensed application crashed if the trace level was set to 7 (VLS_TRACE_ALL) and a custom locking criterion was used. The problem is corrected in this release. The custom locking criterion can be used with the VLS_TRACE_ALL level.
83971	LSRequest, LSUpdate, and LSRelease return VLS_RESOURCE_LOCK_FAILURE. The information is now included in the corresponding documentation of these APIs in the *Sentinel RMS API Reference Guide*.
83842	When a single license server (non-primary) in a redundant license server pool is up and acting as a leader, the reserved tokens (received from earlier leader server) were not released /reclaimed even when the waiting time period had elapsed. ***Note:*** The waiting time period is calculated using the following formula: (The number of servers in the pool) * (license key-lifetime)
85078	In the API Reference Guide, the documentation of VLScgSetHoldTime contained incorrect description of VLScg_NOLIMIT_STRING. The problem is corrected in this release. As per the corrected documentation VLScg_NOLIMIT_STRING can be used for specifying the maximum value of 900 minutes (and not infinite).
85073	In the API Reference Guide, the documentation of VLScgDecodeLicense lacked on information about a scenario in which the meter key will be required. Additional information has been added as a Note in the description of the VLScgDecodeLicense.

Known Problems

This section contains the known problems identified for the product:

Known Problem	Workaround
While generating a standalone revocation permission ticket the license string is not verified. This may result in a permission ticket having an invalid license string. The license string is decoded only at the time of revocation.	Ensure that the license used for permission ticket generation is valid.
The stand-alone revocation ticket generated in a non-Windows platform is not verified correctly in Windows if the "Lock Code Selector Value" provided at the time of permission ticket generation is 0x001 (ID-PROM).	For permission ticket generation use a "Lock Code Selector Value" that is supported on all platforms.
Under certain scenarios the rlftool tool may corrupt while saving the redundant server's configuration file. The settings specified are lost and the error will be returned only in the next execution.	You can run the rlftool again and save the file back by making necessary modifications.
A redundant license is being requested like any other normal license from the follower servers when it has been removed from the leader server.	None
A permission ticket generated in Windows uses a CRLF sequence, whereas in UNIX it uses only CR or LF. This may create a problem in non-Windows platforms when CRLF is read as two different characters.	Modify the ticket "permission_tkt" generated using following command: `perl -pi -e 's/\r\n/\n/g' permission_tkt`

Compatibility and Upgrade Information

Changes in the Separator used for IP Addresses and Host Names in LSHOST

There is a change in the convention for setting the LSHOST[4] environment variable. Now, ~ (tilde) must be used for separating the license servers (IP addresses and\or host names), instead of : (colon), which was valid till 8.2.x.

An example of the new LSHOST string is shown below:

```
Set LSHOST=192.xxx.xx.xx~192.xxx.xx.xx~hostname1
```

Compatibility Information for Version 13 Licenses

The compatibility information for the version 13 licenses is as follows:

- The v8.4.1 license generation library and tool (lscgen) in Windows and Linux allow generation of version 13 and earlier licenses.
- Similarly, the v8.4.1 licensing libraries, license server, and other tools will support all the versions of licenses, except the following:
  - The version 8.4.1 commuter licenses will not work with the applications licensed using the previous version of the licensing libraries (8.2.2 or earlier) in case the license string length exceeds 2048 bytes.
  - The version 8.4.1 grace licenses (installed on a local system) will not work with the applications licensed using the previous version of the licensing libraries (8.2.2 or earlier). Upgrading the licensing library (integrated) to v 8.4.1 can correct this problem.

4. Used for setting the license server on the client end. The application enumerates through the license servers specified in the variable, and sets the one found first.

Installation Information

This section contains information about installing the Sentinel RMS SDK:

Before You Install...

Please check your system for the following before you begin the installation:

- **Obtain the Serial Number for Your SDK**

  You require a valid serial number provided by SafeNet. The installer prompts you to enter the serial number specific to you.

- **If Installing the SDK in the Same Directory...**

  If you are installing the SDK in the same directory where you installed a previous version of the Sentinel RMS HPUX Itanium SDK, then note the following points:

  - The installer will overwrite the files if the installation location on the system contains a previous version of the HPUX Itanium RMS SDK.
  - The root directory of the new installation will take the same directory structure as that of the previous (installed) SDK.
  - Any new files will be copied on the installation location.
  - Any file in use will not be updated. For example, you must stop the Sentinel RMS License Server if running from the same installation directory. Else, the installer will not upgrade it.

- **Check the Firewall Settings**

  Allow the Sentinel RMS license server through the system firewall.

Tip: Sentinel RMS is self-contained in the installation directory. To uninstall, remove the directory. Do stop the Sentinel RMS license server before uninstalling.

System Requirements

The minimum hardware and software requirements for installing the Sentinel RMS SDK are:

Hardware Requirements	Software Requirements
❑ **Processor** Itanium ❑ **Hard Disk Space** 150 MB free hard disk space ❑ **RAM** 128 MB RAM ❑ **Disk Drive** CD-ROM drive (required only if you received the RMS SDK installer on a CD)	❑ **Operating Systems (Client and Server)** HP-UX v11.3x ❑ **Compiler** cc ver A.06.12 ❑ **Java Runtime Environment** Java Runtime Environment v 1.4 or higher for the Java interface

Installation Instructions

1. Insert the Sentinel RMS CD in the appropriate drive of your computer (assuming that you received the RMS SDK installer on a CD).
2. Now, mount the CD using the mount command.
3. The software is packed in a *.tar* archive. Extract files from the archive as follows:

```
$tar -xvf slm_0841_hpux_combo_cdrom.tar
```

4. Change the directory to where the installation script `install.sh` exists. Type `sh INSTALL.sh` to start the installation.
5. Accept the license agreement to proceed with the installation.
6. You will be required to specify the Sentinel RMS serial number—included with the product packaging—to complete the installation.

Installed Directories

The contents of installation are described in the table below:

Directory or File Name	Description
bin	Contains program files and utilities.
examples	Examples that illustrate the various features of Sentinel RMS.
include	The include files, which provide the prototypes for various Sentinel RMS library functions.
lib	The 32-bit licensing and system initialization libraries.
lib64	The 64-bit licensing and system initialization libraries.
Java	Contains the Java interface and related sample files.
Manuals	Contains the Sentinel RMS documentation.

Notes:

- The samples are for C interface only.
- All libraries are non-reentrant and are compiled using the - pthread platform library.
- The binaries included in this release are built with the default GCC compilation options.

License Revocation FAQs

Question: The procedure to revoke network licenses (available since the 8.4.1 release) differs from the earlier one (available since the 8.0.0 release). Why so? I am also confused about the various versions in which the stand-alone and network license revocation features were introduced.

The network revocation process introduced in v8.4.1 is in synchronization with the stand-alone license revocation process. Similar processes should help you in aligning your business work-flows consistently, regardless of the license type. A set of common APIs is provided for network and standalone revocation.

The table below lists the versions of the RMS SDK in which the revocation features are introduced:

License Revocation Feature	RMS Version	License Version
Older method of network license revocation[a]	Introduced in v8.0.0	Can be used to revoke v10 or later licenses
Stand-alone license revocation	Introduced in v8.2.2	Can be used to revoke v11 or later licenses
New method of network license revocation	Introduced in v8.4.1	

a. Described in Appendix D of the *Sentinel RMS SDK Developer's Guide*.

Question: What is backward compatibility provided for the older network revocation workflow?

Older APIs and revocation tools (like, lsdecode) will continue to work as it is. Additionally, the new extended APIs can be used to perform earlier type of revocations.

Question: Questions: Are there any enhancements in the stand-alone license revocation process?

Yes. Custom defined data can also be included in permission ticket for revocation ticket verification.

Technical Support

If you have questions or need additional assistance, please contact Technical Support using this information:

Customer Connection Center (C3)	
http://c3.safenet-inc.com Existing customers with a Customer Connection Center account can log in to manage incidents, get latest software upgrades and access the complete SafeNet Knowledge Base repository.	
Support and Downloads	
http://www.safenet-inc.com/Support Provides access to knowledge base and quick downloads for various products.	
E-mail-based Support	
support@safenet-inc.com	
Telephone-based Support	
United States	(800) 545-6608, (410) 931-7520
France	0825 341000
Germany	01803 7246269
United Kingdom	0870 7529200, +1 410 931-7520
Australia and New Zealand	+1 410 931-7520
China	(86) 10 8851 9191
India	+1 410 931-7520