Sentinel® RMS

SafeNet®

Document Revision History

Part Number 007-010016-001, Revision E, September 2014

Disclaimer and Copyrights

Confidential Information

The Sentinel RMS SDK is designed to protect your applications from unauthorized use. The less information that unauthorized people have regarding your security system, the greater your protection. It is in your best interest to protect the information herein from access by un-authorized individuals.

Contents

Sentinel RMS v8.6.0 for Solaris Sparc (32 and 64-bit) - Release Notes

About This Document

This document contains details about the new features and installation of the product. 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 details about the new features introduced in this release.

Introduced Vendor-specific Isolated License Managers

You can now provide isolated License Manager to your customers. An isolated License Manager is a customized License Manager that has the following features:

- It can host licenses for a specific vendor only.

- It can coexist with the generic License Manager and other vendor's isolated License Manager on the same host

- The isolated License Manager are to be created by vendors by performing customizations in the vendor-specific isolated License Manager library. This is a new 32-bit static library available at the following path: *<installdir>/<version>/lib/IsolatedServer.*

- A new API VLSenableVendorIsolation is provided for assigning a vendor identifier to the isolated License Manager. Also, the API VLSrequestExt2 has been updated to include the vendor identifier parameter.

For more information about the isolated License Managers, refer to the "Chapter - Isolated License Managers" of the Sentinel RMS SDK Developer's Guide.

Introduced Custom Persistence Device (CPD)

In this release, the Custom Persistence Device (CPD) interface is introduced. Using the CPD, vendors can choose custom devices, locations, or databases to store RMS persistence data. By default, persistence data is managed completely by the RMS components. The CPD interface allows vendors to implement a persistence storage approach of their choice. Some examples of persistence storage options are:

- External SQL database

- Data store in the cloud

- Secure memory area of a hardware key

- Hidden/non-OS disk areas

To support custom persistence device, the following changes are introduced:

- Certain Sentinel RMS libraries are enhanced to expose the CPD Configuration API layer. These functions—in the respective libraries and License Manager—determine whether a CPD handler routine defined by the vendor will be used to store the persistence data as opposed to the RMS-managed default persistence scheme. The below table lists them:

API	Library
sntlSetPersistenceDevice	System Initialization
VLSsetPersistenceDevice	- Licensing

API	Library
	▪ Isolated License Manager
VLScleanSetPersistenceDevice	Persistence Cleaning

- ▪ Addition of CPD interface callback APIs (in the header file *lscpdif.h*). The implementation of interface callback APIs is done by vendors for reading, writing or deleting the persistence data records.

- ▪ A typical implementation of interface callback APIs is defined in the sample program "*cpd_ sqlite.c*".

For more information, refer to the "Chapter: Custom Persistence Device" in the Sentinel RMS SDK Developer's Guide.

RMS Licenses Supported Beyond Year 2038

In this release, the Sentinel RMS licenses and licensing operations have been updated to provide a solution of the "Year 2038" problem.

Prior to this release, Sentinel RMS used 32-bit integer datatype (time_t) to process the time in seconds. Based on that limit, it was not possible to have licenses of validity beyond the year 2038. To resolve this problem, the following changes are included:

- ▪ New datatypes introduced:

  - ▪ A new 64-bit datatype (Time64_T) is used in addition to the existing 32-bit member. The 64-bit time members can carry EPOCH time values beyond the year 2038.

  - ▪ The existing 32-bit time member's data type time_t is also changed to Time_T to decouple the width of datatype from the Microsoft specific time flag _USE_32BIT_ TIME_T.

  For definition of datatypes, refer to the header file *lserv.h*.

- ▪ To maintain backward compatibility, the following APIs are introduced (or updated) to make use of the 64-bit datatypes. Refer to the *lserv.h* for API signatures:

API	Reference in the Sentinel RMS SDK API Reference Guide
VLSgetServInfoExt2	*Chapter 1: Sentinel RMS Licensing Library API (Topic "Getting the License Manager Information")*
VLSgetClientInfoExt2	*Chapter 1: Sentinel RMS Licensing Library API*
VLSgetQueuedClientInfoExt	*Chapter 6: License Queuing API*
VLSupdateQueuedClientNew	*Chapter 6: License Queuing API*

- ▪ Changes done in following data structures to make use of Time64_T and Time_T. Refer to *lserv.h* for the data structure definitions:

Changed Data Structures	Reference in the Sentinel RMS SDK API Reference Guide
VLSlicenseInfo	*Chapter 1: Sentinel RMS Licensing Library API (Topic "The Client Query Functions")*
VLSfeatureInfo	*Chapter 1: Sentinel RMS Licensing Library API (Topic "The Feature Query Functions")*
VLSclientInfo	*Chapter 1: Sentinel RMS Licensing Library API (Topic "The Client Query Functions")*
VLSservInfo	*Chapter 1: Sentinel RMS Licensing Library API (Topic "Getting the License Manager Information")*
VLStimeTamperInfo	*Chapter 1: Sentinel RMS Licensing Library API (Topic "Getting the License Manager Information")*
VLSqueuedClientInfo	*Chapter 6: License Queuing API (Topic "VLSqueuedClientInfo Struct")*

- A new structure VLStimeTamperInfo64 is introduced with the time data members of type Time64_T. This is a 64- bit version of the existing structure VLStimeTamperInfo. For more details about the VLStimeTamperInfo64 structure, refer to the topic "Getting the License Manager Information" of the "Chapter 1: Sentinel RMS Licensing Library API" in the Sentinel RMS SDK API Reference Guide.

- All tools and utilities have been updated to support time beyond year 2038.

Introduced Version 15 Licenses

A higher version of RMS licenses—version 15—is introduced in this release. This is because the PIII processor ID (a criteria supported on Linux-based systems only) is no longer supported for license locking purposes.

Allowed Retrieval of Client Specific Log Comments

Earlier, the retrieval of log comments passed using the request, update or checkout APIs was not supported. In this release, the client_log_comment member of the VLSclientInfo structure is being used for retrieving the client specific log comments.

You can use the following APIs for retrieving the log comments: VLSgetClientInfo, VLSgetClientInfoExt, and VLSgetClientInfoExt2.

For more information, refer to the "Chapter 1 - Sentinel RMS Licensing Library API" of the *Sentinel RMS SDK API Reference Guide*.

IPv6 Support Added

Sentinel RMS for Solaris now supports IPv6 (Internet Protocol Version 6) based IP addresses for both client and the RMS License Manager. For detailed information about the new feature, see Appendix A - IPv6 Related Information.

License Manager Customization Enhancements

Apart from the introduction of the isolated License Managers, the following server-side customization enhancements are also included in this release:

Vendor-specified License Manager Cleanup - VLSserverVendorFinalize

A new server-side customization API, VLSserverVendorFinalize, is introduced for performing vendor-specified License Manager cleanup activities right before the License Manager shutdown. For more details, refer to the section "Vendor-specified License Manager Cleanup" of Appendix B - Customization Features, in the Sentinel RMS SDK API Reference Guide.

Vendor-specified License Installing/Loading on License Manager - VLSpolicyLicenseInstall

A new server-side customization API, VLSpolicyLicenseInstall, is provided for customizing the installation\loading of licenses on the License Manager. For more details, refer to the section "Vendor-specified License Installing/Loading on License Manager" of Appendix B - Customization Features, in the Sentinel RMS SDK API Reference Guide.

License Manager Hooking Provided for Pre/post Update Events

Earlier, License Manager customization options were provided only for license request and release. Now onward, you can also add pre and post update hooks by customizing the License Manager. These events will be triggered using the new update API VLSupdateExt.

For more details, refer to the section "Installing Hooks on Pre/Post Request, Update, and Release" of Appendix B - Customization Features, in the Sentinel RMS SDK API Reference Guide.

Introduced a Lighter Version of the VLSgetServInfo API

A lighter variant API—VLSgetServInfoExt2—is introduced to retrieve only requested information instead of obtaining machine fingerprints on each call.

For more details, refer to the section "Getting the License Manager Information" of Chapter 1 - Sentinel RMS Licensing Library API in the Sentinel RMS SDK API Reference Guide.

Introduced Persistence Cleaning Library for 64-bit Platforms (SR-153600)

The persistence cleaning library is now also available for Solaris 64-bit platforms. It contains the VLScleanStandalonePersistenceInfo and VLScleanNetworkPersistenceInfo API functions. Refer to "Chapter - Persistence Cleaning API" of the Sentinel RMS SDK API Reference Guide for details about the API functions.

The new library can be found in the /lib64 directory of the RMS SDK installation.

New Client Library for Creating and Managing Redundant License Configuration File

In this release, a new client library is introduced for creating and managing the redundant license configuration file. The API set provides functionality akin to the rlftool utility. For more details, refer to:

- The "Chapter - Redundant License Configuration File API" of the *Sentinel RMS SDK API Reference Guide* for details about the API functions.

- The "Appendix - Libraries and Compilers Supported" of the Sentinel RMS SDK Developer's Guide for library names and compilers supported.

Usage Log File Related Enhancements

The following usage log file related enhancements are included in this release:

Introduced Log Comment Anonymization

The log comment data inserted by the vendor in various RMS APIs can be anonymized and represented as SHA-256 based unique hash in the usage logs using the anonymous logging option[1]:

This is a two-way process:

- Vendor-end - The data needs to be pre-fixed by the <an> tag. The data following the <an> tag only will be shown anonymized in the usage logs.

- System Administrator-end - The system administrator needs to set up RMS License Manager to allow creation of anonymous usage logs using any of these options:

  - Using the LS_LOG_ANONYMITY Environment Variable

  - Using the `-log-anonymity` License Manager start-up option

To map the anonymous log comment data with the real values, the XML map file is updated (in the similar way as the real host and user names are mapped). Its location is same as before, having the same name as the usage log file and the .xml extension.

Inclusion of Log Comments in the Usage Log Entries

Earlier, only the application-invoked log entries (called as explicit entries) included log comments, if any inserted by your vendor in the API calls. Now, all the entries—including the License Manager generated entries (called as implicit entries)—in usage log will include log comments. For more details, refer to:

- Refer to the Sentinel RMS SDK API Reference Guide for the specifications about log comment data anonymization and APIs that allow log comment data. You may search for "log_comment" keyword to find the APIs that allow log comment.

- Refer to the Sentinel RMS SDK System Administrator's Guide for details on how to configure the License Manager start up options and transaction IDs.

Introduced Additional Transaction IDs in Usage Log

The License Manager usage log categorizes the various activities into transaction types (such as 0 transaction ID when a license is issued, 1 when a license is denied and so on). Earlier, 0 to 9

---

[1]The anonymous logging option was introduced in the 8.5.1 Solaris release for anonymizing the user and host name information of the usage logs.

transaction IDs were used in the usage log to represent the all the transaction types. In this release, a new License Manager startup option -extended- log is introduced, which assigns additional transaction IDs—between 10 to 16—for specific cases in the usage log. Refer to the Sentinel RMS SDK System Administrator's Guide for details about the various transaction IDs.

Increased Limit of the Number of Usage Log Backup Files

The maximum number of backup files allowed is increased from 100 to 10000 for the License Manager usage log files. So, for example, if the log file name is *lserv.log* then the first usage log files will be *lserv.log.00* and the last usage log file in the series will be *lserv.log.9999*.

Introduced Usage Log Backup File Creation API

A new client-side API VLScreateUsageLogBackupFile is provided to allow creation of usage log backup file at vendor's discretion. This API can be used as an alternative to the -z command-line License Manager configuration option available to the system administrator. For more details, refer to the section "Chapter 9 - Usage Log Functions" of the Sentinel RMS SDK API Reference Guide.

Extended Usage Log File Enhancements

The following extended usage log file (generated using the License Manager start-up option **-extended- log**) related enhancements are included in this release:

- Introduced Additional Transaction IDs - In this release, the extended usage log is enhanced to assign additional transaction IDs 17 and 18 for specific cases related to the request and release of a shared license token. Refer to the Sentinel RMS SDK System Administrator's Guide for details about the transaction IDs.

- Client Request Identification - In this release, the extended usage log includes a Client Request identifier (a 16-byte value) that uniquely identifies each client request. Using this identifier, each request log entry can be mapped with the corresponding release log entry.

Refer to the *Sentinel RMS SDK System Administrator's Guide* for details about the usage log file related information, including these enhancements.

License Revocation Related Enhancements

The following license revocation related enhancements are included in this release:

License String Based Network License Revocation

Till now, network license revocation was limited to only the active licenses loaded for a feature-version. In this release, revocation of specific licenses for a feature-version is introduced. This is useful for revoking specific licenses, including:

- Specific licenses on a feature node consisting of multiple license (such as aggregate and additive licenses)

- Active and Inactive licenses

- Future date aggregate licenses

- Licenses not installed yet

For more details, refer to the following topics in the "Chapter - License Revocation" of the Sentinel RMS SDK Developer's Guide:

- Timeline of License Revocation Methods Available in RMS

- Feature Based Revocation versus License String Based Revocation

Deferred License Revocation

It is now possible to incorporate a grace period with license revocation. This allows a revoke operation to be scheduled to some future point in time instead of performing the revoke operation in real-time. The grace period, known as the deferred revocation grace period, is defined by the vendor at the time of generating the permission ticket. It can be a value between 0 days (revocation happens instantaneously) and 30 days (revocation happens 30 days after it was actually instigated).

For more details, refer to the topic "Deferred Revocation" in "Chapter - License Revocation" of the Sentinel RMS SDK Developer's Guide.

Support for Revocation of Redundant Licenses

In this release, revocation of redundancy-enabled network licenses is provided. The detailed workflow is described in the "Chapter - Redundant License Managers" of the Sentinel RMS SDK Developer's Guide.

Enhancement in the Persistence Cleaning API (SRs 156456, 156462, and 141721)

Inconsistent behavior was observed in using the persistence cleaning APIs— VLScleanStandalonePersistenceInfo and VLScleanNetworkPersistenceInfo. For example, the time tampering persistence for licensing library versions 8, v8.2 and v8.4 licenses was successfully cleaned using the VLScleanStandalonePersistenceInfo API. However, the operation failed for version 7, 8.3 and 8.5 licenses using the same API.

To correct this problem, changes have been made in the persistence cleaning APIs' design so that now a user can pass the actual version as the first parameter (clientLibVersion and serverVersion for the VLScleanStandalonePersistenceInfo and VLScleanNetworkPersistenceInfo API respectively).

Refer to the "Chapter - Persistence Cleaning API" in the Sentinel RMS SDK API Reference Guide for details.

Enhancement in Obtaining Ethernet based Locking Codes (SR 148294)

RMS scans all the network interfaces (physical, loopback, and P2P) for supporting Ethernet based locking. However, it allows license locking only to physical Ethernet interfaces. Earlier, RMS scanned up to 10 entries on a system which caused locking code generation to fail if any of these entries did not have a physical interface. In this release, this limit has been raised to 30, hence increasing the scope of successful locking code generation. This enhancement is applicable to the

following API functions: VLSgetMachineID, VLSgetNumberedMachineID, and VLSgetNumberedMachineIDExt.

Note the following exceptions regarding this enhancement:

- This attempt may still not provide desired results under certain scenarios. For example, if the system has 30 IPs assigned to loopback interface, then locking code generation will fail if the physical Ethernet interface is located beyond 30 interfaces.

- The locking code will not be generated if:
    - The interface is not assigned an IPv4 address.
    - If the network interface is down.

- In the case of VLSgetNumberedMachineID and VLSgetNumberedMachineIDExt API functions, a change in index may occur if the loopback interface not assigned an loopback address.

Problems Corrected in This Release

In this release, the following problems are corrected:

US \ SR #	Problem Description
152600 and 161883	Automatic license renewal related fixes
173169 and 178204	Failure to release license tokens after license request exceeded 32,800 tokens
152603 and 157897	Error 18 not shown for expired licenses
1354	Memory leak problem with VLSgetLeaderServerName API
153789 and 163088	Memory leak problem with Volume Transaction License APIs
RMS-3028	License registration problem
RMS-2318	Enhancements in the persistence data module.
RMS-1728	Performance degradation - unusually high number of page faults in RMS Library
RMS-2552	Valgrind warnings in Virtual Machine detection module
RMS-2603	Some API calls returning VLS_RESOURCE_LOCK_FAILURE
RMS-2040	Documentation update - better distinction between Windows and non-Windows
RMS-2754	Confusing Developer's/API Guide documentation about Autotimer
RMS-2755	Documentation incorrect about Server Port Change
RMS-1332	VLSgeneratePermissionTicketExt() documentation error
RMS-3531	Invalid license string error while rebuilding the License Decoding Utility with license encryption
RMS-3530	Legacy issues with vendor encrypted license codes
RMS-3529	When you rebuild the License Manger and load an encrypted license file, the License Manager crashes.
RMS-3430	Compatibility issues among RMS 8.2 and RMS 8.5
RMS-2945	Crash in revocation tool while decoding the revocation ticket (using the Sentinel Caffe API VLScgDecodeLicenseRevocationTicketExt)
RMS-2944	Issue in VLSusageAuthenticate API related to MappinXML file
RMS-2937	Crash in VLSusageAuthenticate , VLSusageFileDecrypt and vusage tools while parsing a usage log file that contains log comments with six or more anonymized words.
RMS-2641	Enhancement request in Group feature attribute
RMS-2640	VLSgetNumberedMachineID API crashes when called 5 times.
RMS-2637	Increase license limit from 1000 to 2000
RMS-2611	The LSHOST value is getting Reset
RMS-2258	The VLSusageFileDecrypt API does not provide Session ID in CSV file

Installation Information

This section contains following information about installation of the product:

- "Before You Install..." on page 16
- "System Requirements" on page 17
- "Installation Instructions" on page 18
- "Installed Directories" on page 19

Before You Install...

Before you begin installing the RMS SDK, note the following:

Administrator Privileges

You must have Administrator privileges to successfully install all the components.

RMS SDK Serial Number

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

Check the Firewall Settings

Allow the Sentinel RMS License Manager through the system firewall. You will be prompted for this during the installation.

System Requirements

To use the Sentinel RMS, the minimum hardware and software requirements are:

Hardware Requirements	Software Requirements
Processors: Sparc	**Operating Systems:** For 32-bit applications: Solaris Sparc 5.10, 5.11 For 64-bit applications: Solaris Sparc-64 5.10, 5.11
Monitor and Settings: VGA monitor with 800 x 600 resolution (1024 x 768 recommended)	**Compilers:** GCC Compiler v3.4.3 (32-bit) GCC Compiler v3.4.3 (64-bit) Libc v 1.1 (32-bit, 64-bit)
Hard Disk Space: 100 MB free hard disk space	**Java Runtime Environment**: Java Runtime Environment v1.5 for the Java interface
RAM: 128 MB RAM	**Web Browser (for viewing HTML Helps):** Firefox 3.5 or later
Peripherals: ■ CD\DVD-ROM drive (required only if you received the RMS SDK installer on a CD\DVD) ■ A USB port or a parallel port to attach the license meter key. This is required on the system where licenses are to be generated.	**PDF File Viewer (for viewing PDF documents):** Adobe Acrobat 4.0 (or later)

Installation Instructions

1. Insert the Sentinel RMS media in the appropriate drive of your computer.

2. Now, mount it using the mount command.

3. The Software is packed in a .tar archive . Extract files from the archive as follows:
    `tar -xvf slm_0860_sol2sprc_combo_cdrom.tar`

4. Change the directory where the installation script *install.sh* exists. Afterward, 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 RMS SDK serial number-included with the product packaging to complete the installation.

7. You will be required to choose whether you want to allow the License Manager to start automatically on system reboot. If you allow this configuration, you will be required to reboot. However, if you do not allow this configuration, you will need to manually start the License Manager (from the /bin directory) each time you reboot the system.

Sentinel RMS is self-contained in the installation directory. To uninstall, remove the directory. Do stop the Sentinel RMS License Manager before uninstalling.

Installed Directories

The following directories are installed:

Directory or File	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.
ReleaseNotes.pdf	This file.

Documentation Resources

This section provides details about the additional documentation resources for the product.

Developer's Guide

Contains the complete product overview, the necessary information for licensing and distributing the applications. Useful for developers planning and implementing licensing. To access, use the following ways:

- PDF and HTML versions installed on the system can be accessed using the Doc Access Page (*.htm*) in the *Manuals* directory.

- The latest online copy can be accessed here:
  http://sentinelrms.safenet-inc.com/RMSDocumentation/Vendor/Default.htm

API Reference Guide

Contains details about all the API functions, including the licensing library, license code generator, system initialization, and so on. Useful for developers integrating the API functions in the code. To access, use the following ways:

- PDF and HTML versions installed on the system can be accessed using the Doc Access Page (*.htm*) in the *Manuals* directory.

- The latest online copy can be accessed here:
  http://sentinelrms.safenet-inc.com/RMSDocumentation/Vendor/Default.htm

> The Developer's Guide and API Reference Guide are bundled together as the Vendor's documentation. You can find it as the last item in its ToC (in the left-side pane).

System Administrator's Guide

Contains details about using the system administration and License Manager configuration options. Useful for the System Administrator of the end user (on the customer site). To access, use the following ways:

- The HTML version installed on the system can be accessed using the Doc Access Page (*.htm*) in the *Manuals* directory.

- The latest online copy can be accessed here:
  http://sentinelrms.safenet-inc.com/RMSDocumentation/SysAdmin/Default.htm

Appendix A - IPv6 Related Information

Here are the detailed highlights of the IPv6 feature included in this release:

New Environment Variable to Set the Client-side Protocol

A client can contact the License Manager using the IPv4 or IPv6 communication protocol. A new client-side environment variable, LSTCPIPVER, is provided to set the protocol (IPv4 is the default protocol setting).

> The communication protocol cannot be switched (from IPv6 to IPv4, and vice versa) dynamically between the VLSinitialize and VLScleanup calls in an application.

When the License Manager is hosted on a dual stack system, both IPv4 and IPv6 clients can communicate with it. Take a look at the various License Manager and client configurations:

License Manager Host Protocol	Client-side Protocol	
	IPv4	IPv6
IPv4	Supported	Not supported
IPv6	Not supported	Supported
Dual stack	Supported	Supported

IPv6 and Redundant Server Pool

It is possible to create redundant License Manager pool consisting of IPv6 based systems. However, the redundant License Manager configuration file (by default named lservrlf) can store either IPv6 or IPv4 addresses i.e. a redundant License Manager pool consisting of both the License Managers (both IPv4 and IPv6) is not supported. So, if the pool consists of IPv6 systems, only IPv6 based clients can communicate in the setup. The similar behavior is true for IPv4 setup.

IPv6 and Commuter Licensing

The combinations, shown in the table above, hold true in the case of commuter licensing as well:

- If the License Manager is set up on a dual stack machine, then both IPv4 and IPv6 clients can checkout a commuter token from it.

- If the License Manager is set up in pure IPv6 environment, then only IPv6 clients can checkout a commuter token.

- If the License Manager is set up in pure IPv4 environment, then only IPv4 clients can checkout a commuter token.

> The commuter token can be checked in only in the mode in which it was checked out even if the License Manager is set up on a dual stack machine. The commuter token can be checked-in only using the same IP Address with which it was checked out.

IPv6 and Group Reservation

A group reservation file can contain both IPv4 and/or IPv6 addresses. However, if a client is communicating using IPv4, the License Manager will consider IPv4 addresses only. Similarly, if client is communicating using IPv6, the License Manager will consider only IPv6 address.

Known Problems to the IPv6 Support

Here are the known problems related to the IPv6 support:

- License locking for IPv6 addresses is not supported. Hence, the locking code generators (like, echoid and rcommute) cannot generate IPv6 based locking codes even if the LSTCPIPVER environment variable is set to 6 (means IPv6). In the case of a dual stack system, IPv4 locked licenses will be supported in IPv4 as well as IPv6 mode.

- Redundant License Manager pool is not supported when Link Local addresses of the License Managers are used.

- Checkin of a commuted token fails when Link Local address of the License Manager is used.

- Site (subnet) licensing is not supported with IPv6 addresses.

Technical Support

You can contact us using any of the following options:

- **Business Contacts** - To find the nearest office or distributor, use the following URL:
  http://www.safenet-inc.com/contact-us/

- **Technical Support** - To obtain assistance in using SafeNet products, feel free to contact our Technical Support team:

  - Phone: 800-545-6608 (US toll free), +1-410-931-7520 (International)

  - E-mail: support@safenet-inc.com

  - URL: http://sentinelcustomer.safenet-inc.com/sentinelsupport/

- **Downloads** - You may want to check out updated installers and other components here:
  www.sentinelcustomer.safenet-inc.com/sentineldownloads/