

# Sentinel LDK 10.0

## SOFTWARE PROTECTION AND LICENSING GUIDE



## Revision History

Part number 007-002031-001, Revision A, 2306-1

## Disclaimer and Copyrights

All information herein is either public information or is the property of and owned solely by Thales and/or its subsidiaries and affiliates who shall have and keep the sole right to file patent applications or any other kind of intellectual property protection in connection with such information.

Nothing herein shall be construed as implying or granting to you any rights, by license, grant or otherwise, under any intellectual and/or industrial property rights of or concerning any of Thales and any of its subsidiaries and affiliates (collectively referred to herein after as “Thales”) information.

This document can be used for informational, non-commercial, internal and personal use only provided that:

- The copyright notice below, the confidentiality and proprietary legend and this full warning notice appear in all copies.
- This document shall not be posted on any publicly accessible network computer or broadcast in any media and no modification of any part of this document shall be made.

Use for any other purpose is expressly prohibited and may result in severe civil and criminal liabilities.

The information contained in this document is provided "AS IS" without any warranty of any kind. Unless otherwise expressly agreed in writing, Thales makes no warranty as to the value or accuracy of information contained herein. The document could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Furthermore, Thales reserves the right to make any change or improvement in the specifications data, information, and the like described herein, at any time.

**Thales hereby disclaims all warranties and conditions with regard to the information contained herein, including all implied warranties of merchantability, fitness for a particular purpose, title and non-infringement. In no event shall Thales be liable, whether in contract, tort or otherwise, for any indirect, special or consequential damages or any damages whatsoever including but not limited to damages resulting from loss of use, data, profits, revenues, or customers, arising out of or in connection with the use or performance of information contained in this document.**

**Thales does not and shall not warrant that this product will be resistant to all possible attacks and shall not incur, and disclaims, any liability in this respect. Even if each product is compliant with current security standards in force on the date of their design, security mechanisms' resistance necessarily evolves according to the state of the art in security and notably under the emergence of new attacks. Under no circumstances, shall Thales be held liable for any third party actions and in particular in case of any successful attack against systems or equipment incorporating Thales products. Thales disclaims any liability with respect to security for direct, indirect, incidental or consequential damages that result from any use of its products. It is further stressed that independent testing and verification by the person using the product is particularly encouraged, especially in any application in which defective, incorrect or insecure functioning could result in damage to persons or property, denial of service or loss of privacy.**

© 2024 THALES. All rights reserved. Thales, the Thales logo and Sentinel are trademarks and service marks of Thales and/or its subsidiaries and affiliates and are registered in certain countries. All other trademarks and service marks, whether registered or not in specific countries, are the properties of their respective owners.

# CONTENTS

<b>Sentinel LDK Software Protection and Licensing Guide</b>	<b>9</b>
Contents of the Sentinel License Development Kit	9
Sentinel LDK - Demo Kit	9
Sentinel LDK - Starter Kit	9
About This Guide	10
Major Components of the Vendor Suite	10
Sentinel LDK Envelope and Sentinel Licensing API	10
Sentinel LDK Data Protection Utility	11
Sentinel LDK ToolBox	11
Admin Control Center	11
Sentinel RUS (Remote Update System)	12
Sentinel LDK-EMS (Entitlement Management System)	12
Sentinel License Generation API	12
Sentinel LDK-EMS Customer Portal	12
Master Wizard	13
Migrating to Sentinel LDK From Other Systems	13
<b>Part 1 - Getting Started With Sentinel LDK Vendor Suite</b>	<b>14</b>
Chapter 1: Understanding Sentinel LDK Software Protection and Licensing	15
Fundamentals of Protection	15
Major Protection Solutions	16
Fundamentals of Licensing	18
Flexible and Secure Licensing Solutions	18
Principles of Sentinel LDK	20
Customizing Your Unique Solution	21
Sentinel HL Keys	26
Benefits of Sentinel HL Key Protection	27
Sentinel SL Keys	28
Benefits of Sentinel SL Key Protection	28
Sentinel SL Unlocked Licenses	29
Obtaining Additional Information About Sentinel LDK	32
Chapter 2: Understanding Cloud Licensing	33
Overview	33
Implementing Cloud Licensing	33
Benefits of Cloud Licensing	34
<b>Part 2 - Protection</b>	<b>36</b>
Chapter 3: Protecting Software	37
Sentinel LDK Protection	37
Elements of Sentinel LDK Protection	38

Selecting a Protection Method .....	41
Chapter 4: Sentinel Licensing API Protection .....	43
Sentinel Licensing API Overview .....	43
Sentinel Licensing API Prerequisites .....	44
Learning About the Sentinel Licensing API .....	46
Sentinel Licensing API Implementation .....	47
Sentinel Licensing API Functionality .....	50
Chapter 5: Sentinel LDK Envelope Protection .....	52
Functionality .....	52
Sentinel LDK Envelope for Windows .....	56
Protecting .NET Assemblies .....	62
Protecting Python Applications .....	66
Sentinel LDK Envelope for Linux Applications .....	67
Sentinel LDK Envelope for Mac Binaries .....	67
Sentinel LDK Envelope for Java Executables .....	69
Chapter 6: Protection Strategies .....	72
Protection Strategies Overview .....	72
General Protection Guidelines .....	73
Types of Attack and Their Sentinel LDK Defense .....	73
Chapter 7: Protecting Data Files .....	77
Overview .....	77
Data File Protection Prerequisites .....	80
Launching Sentinel LDK Data Protection Utility .....	81
Licensing Data Files—Getting Started .....	82
Working With the dfcrypt Command Line Utility .....	83

## Part 3 - Licensing ..... 88

Chapter 8: Introduction to Sentinel LDK-EMS .....	90
Sentinel LDK-EMS Overview .....	90
User Types and User Roles in Sentinel LDK-EMS .....	92
Getting Started With Sentinel LDK-EMS .....	95
Sentinel License Generation API .....	96
Switching Between Back-ends to Maintain Protection Keys .....	96
Chapter 9: Preparing Your Sentinel LDK Licensing Plan .....	98
Licensing Overview .....	98
Preparing Your Licensing Plan .....	99
Choosing the Protection Level for Your Products .....	101
Designating Products for Trial or Grace Period Use .....	104
Assigning License Terms to Features .....	104
Utilizing Protection Key Memory .....	106
Using Your Licensing Plan With Sentinel LDK-EMS .....	107
Chapter 10: Implementing Your Sentinel LDK Licensing Plan .....	108
License Planning in Sentinel LDK-EMS .....	108
Managing Features .....	109
Managing Products .....	110
Maintaining Products and Licenses .....	118
Chapter 11: Sentinel LDK Entitlements, Production, and Development Tasks .....	122

Sentinel LDK Entitlement Processing and Production .....	122
Managing Entitlements .....	123
Producing Entitlements .....	131
Withdrawing and Deleting Entitlements .....	132
Customer Portal - Activating Entitlements .....	133
Customer Portal - Applying Updates to Protection Keys .....	133
Viewing License Updates .....	134
Applying License Updates to SL AdminMode Keys .....	135
Performing Development-related Tasks .....	136
Enabling Trial Use and Grace Periods .....	140
Chapter 12: Sentinel LDK Administration and Customer Services .....	142
Administration Tasks .....	142
Customer Services .....	145
Channel Partners .....	145
Chapter 13: Sentinel Remote Update System (RUS) .....	147
RUS Utility Overview .....	147
RUS Workflow .....	148
Using the RUS utility .....	149
Chapter 14: Generating Sentinel LDK Reports .....	153
Managing Reports .....	153
Permissions for Working With Reports .....	154
Scheduling Reports .....	154
Presentation Formats .....	154
Export Formats .....	154
Available Reports .....	154
Custom Reports .....	155
Chapter 15: Cloud Licensing Using Sentinel LDK Cloud Portal .....	156
Cloud Licensing Overview .....	156
Implementation Summary .....	158
Setting Up the License Server Machine .....	159
Installing a Client Identity on an End User's Machine .....	160
Cloud Licensing Performance .....	161
Overview for Multi-Level License Servers .....	162
Online Detach: Setting Up Multi-Level License Server Machines .....	165
Offline Detach: Setting Up Multi-Level License Server Machines .....	167
Configuring High Availability for Cloud Licensing .....	170
Chapter 16: Cloud Licensing Using Sentinel Admin Control Center .....	171
Cloud Licensing Overview .....	171
Vendor-Hosted Implementation Summary .....	173
Setting Up the License Server Machine .....	174
Installing a Client Identity on an End User's Machine .....	176
Customer-Hosted Implementation Summary .....	178
Cloud Licensing Performance .....	178
Overview for Multi-Level License Servers .....	179
Online Detach: Setting Up Multi-Level License Server Machines .....	182
Offline Detach: Setting Up Multi-Level License Server Machines .....	184
Configuring High Availability for Cloud Licensing .....	187

<b>Part 4 - Distributing Software</b>	<b>188</b>
Chapter 17: Distributing Sentinel LDK With Your Software	189
Sentinel LDK Software for End Users	189
Distributing Sentinel LDK Run-time Environment	193
Chapter 18: Sentinel License Manager	208
Types of License Managers	209
Selection of the License Manager By the Protected Application	211
License Manager Tools	214
Managing User Access to Admin License Manager Information	215
Managing Access to Standalone and Network Licenses	217
Returning Network Seats to an SL License	220
Working Directly With License Manager Configuration Files	221
Configuring Detachable License Definitions	226
Making Product Names Visible on the End User's Machine	227
Loss of Connection With a Network License	228
Chapter 19: Sentinel Admin Control Center	229
Launching Admin Control Center	230
Admin Control Center Interface	230
Display of Protection Keys and Sessions in Admin Control Center	232
Administrator's Workflow	233
Configuration Considerations	233
Diagnostics	234
Customizing Admin Control Center Look and Feel	235
<b>Part 5 - Licensing Business Models</b>	<b>239</b>
Chapter 20: Sentinel LDK Licensing Business Models: Overview	240
Introduction	240
Sentinel LDK Licensing	241
Determining the Best Protection and Licensing Method	242
Chapter 21: Sentinel LDK Licensing Business Models: Description of Models	243
Evaluation Licensing Business Models	245
Component-based Licensing Business Models	253
Metered Licensing Business Models	257
Locked Licensing Business Models	274
Mobile Licensing Business Models	279
Network Licensing Business Models	283
Sales Boosting Licensing Business Models	293
Perpetual Licensing Business Models	300
<b>Part 6 - Appendices</b>	<b>303</b>
Appendix A: Understanding the Sentinel LDK Master License	305
What is the Master License?	306
Where is the Master License Located?	306
Updating the Master License	307
Modules Summary	307
Trial Licenses Provided With Sentinel LDK	308
Licensing Concepts	309

Product Activation Module .....	310
New SL Key Pool .....	311
Network Seats .....	311
Unlocked Trialware Module .....	315
Unlocked Unlimited Module .....	315
V-Clock Module .....	316
AppOnChip Module .....	316
Channel Partner Module .....	317
Reporting Module .....	317
Cloud Licensing Module .....	317
Appendix B: Sentinel LDK Run-time Network Activity .....	319
Local Communications .....	319
Remote Communications .....	320
Appendix C: Maximum Number of Features in a Sentinel HL Key .....	322
Appendix D: How Sentinel LDK Detects Machine Cloning .....	323
How Clone Protection Schemes Work .....	323
Using the "Platform Default" Scheme .....	324
Summary of Clone Protection Schemes .....	327
Requirements for Each Clone Protection Scheme .....	328
Clone Detection for Physical Machines .....	330
Clone Detection for Virtual Machines .....	334
How to Analyze a Clone Report .....	340
How to Clear the "Cloned" Status for a Product License .....	345
Appendix E: How Sentinel LDK Protects Time-based Licenses With V-Clock .....	346
Tampering with the System Clock .....	347
Re-enabling a Blocked Protected Application .....	347
Setting Fallback to V-Clock If the RTC Battery in a Sentinel HL key is Depleted .....	348
Appendix F: How to Optimize Performance for Sentinel LDK Run-time Environment .....	349
SL UserMode License .....	349
Run-time Environment .....	349
Testing for Presence of Features .....	349
Appendix G: Upgrading Sentinel HL Keys .....	350
Upgrading a Sentinel HL Key to Driverless Configuration .....	350
Converting a Sentinel HL Standalone Key to a Network Key .....	353
Differences Between Sentinel HL (Driverless configuration) keys and Sentinel HASP keys .....	354
Appendix H: Protecting Applications in Docker Containers .....	355
Using SL Keys .....	356
Using HL Keys .....	358
Additional Considerations .....	359
Side-By-Side Comparison .....	360
Appendix I: Protecting Applications in Linux LXC Containers .....	362
Using SL Keys .....	363
Using HL Keys .....	364
Appendix J: Troubleshooting .....	366
Checklist .....	366
Problems and Solutions .....	366
Appendix K: Choosing Between Sentinel LDK-EMS and Sentinel EMS .....	371

Glossary .....372



# Sentinel LDK Software Protection and Licensing Guide

This section provides an introduction to Sentinel Vendor Suite. Thales recommends that you review this information to familiarize yourself with:

- > The contents of the Sentinel License Development Kit – Starter or Demo kit
- > The major components of Sentinel Vendor Suite
- > The information provided in this guide
- > How to obtain additional technical support for these products

## Contents of the Sentinel License Development Kit

The two Sentinel License Development Kits (*Sentinel LDK*) described below are available as part of the Sentinel Vendor Suite.

### Sentinel LDK - Demo Kit

The Sentinel License Development Kit - Demo kit contains the software and hardware you need to evaluate Sentinel LDK protection and licensing. The following items are included:

- > Links to the documentation and installation software for Sentinel LDK
- > Sentinel HL Demo keys to facilitate the evaluation process
- > Sentinel LDK Software Protection and Licensing Quick Start card

The most up-to-date documentation can also be found at [Sentinel LDK documentation](#).

### Sentinel LDK - Starter Kit

The Sentinel License Development Kit - Starter kit contains the software and hardware you need to apply Sentinel LDK protection and licensing. The following items are included:

- > Links to the documentation and installation software for Sentinel LDK
- > Sentinel LDK Vendor keys:
  - Developer key for applying protection
  - Master key for generating license updates (only used with Sentinel LDK-EMS installed on-premises)
- > Sentinel LDK Software Protection and Licensing Quick Start card

The most up-to-date documentation can also be found at [Sentinel LDK documentation](#).

Sentinel HL keys for distribution to your customers must be ordered separately.

## About This Guide

---

This guide is designed to help software publishers protect and license their software using Sentinel LDK. The guide provides background information and details about how Sentinel LDK can best serve your protection and licensing requirements.

The guide is divided into the following parts:

- > **"Part 1 - Getting Started With Sentinel LDK Vendor Suite " on page 14** – Introduces Sentinel LDK, presents basic protection and licensing concepts, and leads you through the process of configuring the system. You should read this part after opening your kit.
- > **"Part 2 - Protection" on page 36** – Provides an in-depth presentation of Sentinel LDK protection methods. This part includes strategies for maximizing the protection of your software using Sentinel LDK. This part is specifically for software engineers who have the responsibility for using the Sentinel LDK protection applications to protect software.
- > **"Part 3 - Licensing" on page 88** – Discusses the options that Sentinel LDK provides to enable you to apply flexible licensing terms to your software and provides case studies for you to examine. This part is particularly relevant to product and business managers who have to make decisions about how their software is licensed. This part should also be read by operations staff and others involved in production.
- > **"Part 4 - Distributing Software" on page 188** – Details the Sentinel LDK software that can be delivered to end users to ensure easy and trouble-free deployment of protected software. This part also describes the various ways of effectively delivering the Sentinel LDK software components.
- > **"Part 5 - Licensing Business Models" on page 239** – Provides an overview and detailed description of the various Sentinel LDK Licensing business models that you can use to distribute your software.
- > **"Part 6 - Appendices" on page 303** – Provides supplementary information regarding Sentinel LDK.
- > **"Glossary" on page 372**

## Major Components of the Vendor Suite

---

Sentinel License Development Kit (*Sentinel LDK*) Vendor Suite contains many modules, tools, and APIs that assist you to manage the protection and licensing of your application. This section provides an overview of the most significant items in the Vendor Suite.

## Sentinel LDK Envelope and Sentinel Licensing API

---

**Sentinel LDK Envelope** is a tool that wraps your application in a protective shield. This shield ensures that:

- > The application is protected against disassembly and reverse engineering. Your intellectual property is protected.
- > The protected application cannot run unless a suitable Sentinel protection key can be accessed by the application.

An application that has been protected by Sentinel LDK Envelope can contain the **Data File Protection module** to automatically encrypt data files to disk and to read them back. You can use the **Sentinel LDK Data Protection utility** to pre-encrypt data files for use with the protected application.

You can use **Sentinel Licensing API** to provide enhanced protection for your application and to enable the licensing of specific Features in the application.

## Sentinel LDK Data Protection Utility

---

Sentinel LDK Data Protection utility is a tool that can do either of the following:

- > Protect data files with encryption. A protected data file can only be accessed by an application that has been protected with Sentinel LDK Envelope and that possesses the required encryption key.
- > Protect data files with encryption and licensing. A protected data file can be accessed:
  - only by an application that has been protected with Sentinel LDK Envelope AND
  - only when the end user has the required license on a protection key.

## Sentinel LDK ToolBox

---

Sentinel LDK ToolBox is an interactive application that enables software developers to learn about the following Sentinel APIs:

- > Sentinel Licensing API
- > Sentinel License Generation API
- > Sentinel Admin API

In ToolBox, software developers can execute API functions, observe the behavior of the functions, and then copy the relevant source code into their own applications.

## Admin Control Center

---

Sentinel Admin Control Center is a customizable, web-based, end-user utility that enables centralized administration of Sentinel License Managers and Sentinel protection keys.

## Sentinel RUS (Remote Update System)

---

Sentinel RUS utility is an advanced tool that enables you to perform secure, remote updating of the license and memory data of Sentinel protection keys after they have been deployed on the end user's computer.

## Sentinel LDK-EMS (Entitlement Management System)

---

Sentinel LDK-EMS is a web-based graphical application that is used to perform a range of functions required to manage the licensing, distribution, and maintenance of protected applications and data files.

You can use **Sentinel LDK-EMS Web Services** to perform the same functions programmatically. This enables you to integrate the EMS functionality into your own back end infrastructure.

## Sentinel License Generation API

---

For software vendors who prefer to use their own ERP back-ends, Sentinel License Generation API provides access to the power and flexibility of Sentinel protection keys without the need to install the full Sentinel LDK-EMS system. You can use Sentinel LDK ToolBox to examine the API functions, create license templates, and to generate protection keys.

## Sentinel LDK-EMS Customer Portal

---

The Sentinel LDK-EMS Customer Portal is a Web portal that your customer can access to:

- use the online Product activation mechanism. This mechanism enables a customer to:
  - convert a trialware version of your protected application or data file (an *unlocked trialware Product*) to a fully-licensed version (a *Locked Product*).
  - directly activate a Locked Product.

The end users activate the relevant version using a unique Product Key that they receive from you after completing the required commercial transaction to purchase a license for the application. In either case, your investment against software piracy is protected.

The customer logs in to the Customer Portal by providing a Product Key. The customer completes a registration form (if you require this) and then chooses the method to activate the Product.

- Online activation is completely automatic and activates the license on the local machine.
- Offline activation enables the customer to download a utility that can be used to activate the license manually on the same machine or on a different machine.

The Sentinel LDK tutorials leads you through the complete process: define a Feature in Sentinel LDK-EMS, define Products, enter an order, generate a product key, and finally convert the trialware to a Locked Product using the Customer Portal.

- > use the online update mechanism. This mechanism enables a customer to automatically apply all outstanding updates from the vendor to all the protection keys located on the customer's machine.

## Master Wizard

---

You use the Sentinel LDK Master Wizard tool to introduce one of your Vendor keys to Sentinel LDK. This process performs the following:

- > Registers your Batch Code in Sentinel LDK-EMS. This is required so that Sentinel LDK-EMS will recognize the Batch Code and perform actions for that Batch Code.
- > Downloads vendor-specific APIs and the vendor library from Thales servers. These are generated individually for each Batch Code and are used by Sentinel LDK Envelope and Licensing API to implement license enforcement.
- > Downloads Run-time Environment (RTE) installers from Thales servers and customize them on your machine. The Master Wizard customizes the installers by embedding your vendor library and the server address for Sentinel LDK-EMS.

For information on using the Master Wizard to introduce a Vendor key, see the [Sentinel LDK Installation Guide](#).

## Migrating to Sentinel LDK From Other Systems

---

Migrating to Sentinel LDK from other types of protection schemes can be a significant challenge. Thales provides detailed system-specific migration guides to assist you in this transition. Each migration guide provides a number of different multi-stage migration options. These options describe how you can prepare to use Sentinel LDK protection for new customers while you migrate your existing customers to Sentinel LDK at a pace that you can manage.

[Migration guides](#) are provided for the following systems:

- > Hardlock
- > SmartKey
- > Sentinel SuperPro
- > HASP HL
- > HASP4
- > Sentinel Hardware Keys
- > Sentinel HASP

# Part 1 - Getting Started With Sentinel LDK Vendor Suite

## In this section:

---

- > ["Understanding Sentinel LDK Software Protection and Licensing" on page 15](#) – Provides an overview of the concepts of software and intellectual property protection and licensing, discusses the primary protection solutions, and focuses on how Sentinel LDK provides a comprehensive solution to all your protection requirements.
- > ["Understanding Cloud Licensing" on page 33](#) – Provides an alternative for software-based licensing called *cloud licensing*. This model is based on the software -based protection keys, but it provides a simpler mechanism, both for the vendor and for the end user, for distributing and managing licenses.

# CHAPTER 1: Understanding Sentinel LDK Software Protection and Licensing

This section provides an overview of the concepts of software and intellectual property protection and licensing, discusses the primary protection solutions, and focuses on how Sentinel LDK provides a comprehensive solution to all your protection needs.

Thales recommends that you familiarize yourself with the information in this section so that you can maximize the benefits of using Sentinel LDK.

*In this section:*

- > ["Fundamentals of Protection" below](#)
- > ["Major Protection Solutions" on the next page](#)
- > ["Fundamentals of Licensing" on page 18](#)
- > ["Flexible and Secure Licensing Solutions" on page 18](#)
- > ["Principles of Sentinel LDK" on page 20](#)
- > ["Customizing Your Unique Solution" on page 21](#)

## Fundamentals of Protection

This section examines the nature of protection, and identifies the two types of protection that you need to consider.

### What is Protection?

*Protection* is the process of securing an application or intellectual property by incorporating automated and customized security strategies.

Protection is achieved by implementing specific security strategies, such as wrapping your application in a security envelope, and incorporating various security measures within the application's code during development. The greater the number of security measures incorporated, and the higher the level of their complexity, the more secure your application becomes.

It is not sufficient to protect only your software—you must also protect your intellectual property. Your professional expertise and the secrets that you use in developing your software, for example algorithms, must also be protected.

## Copy Protection

Copy protection is the process of encrypting your software and incorporating various security measures throughout the code and binding it to a key so that it can only be accessed by authorized users who are in possession of the key. The more complex the copy protection applied to your software, the less likely it is to be compromised.

Similarly, important data files can be encrypted and protected with licensing so that only users who possess the key are able to access the files.

## Intellectual Property Protection

Your intellectual property is the foundation on which your products are developed. Intellectual property theft is surprisingly easy. Every year, companies report the loss of proprietary information and intellectual property valued at many billions of dollars.

The algorithms and other secret information that you use to make your products unique and competitive must be protected against attempts to discover their secrets, or to apply reverse engineering to the software code.

## Major Protection Solutions

---

With Sentinel LDK, the ability to protect and license your software is facilitated by the use of flexible protection and licensing tools, together with a Sentinel protection key to which your software is subsequently bonded. This key may be either hardware-based or software-based.

**NOTE** In general, references in this section to protection and licensing of software are also applicable to protection and licensing of data files.

### Hardware-based Solutions

In hardware-based solutions, you supply an external hardware device together with your software. The functioning of your software is dependent on the device being connected to the end user's computer. At run-time, your software communicates with the hardware device, and only functions correctly if it receives an authentic response from the device.

Sentinel LDK provides a variety of hardware devices in the form of Sentinel HL keys. You can select the type of Sentinel HL key that best suits your requirements. For more information about Sentinel HL keys, see ["Sentinel HL Keys" on page 26](#).

### Software-based Solutions

In software-based solutions, following the installation of your software on an end user's computer, the protection and licensing is bonded to that specific machine. Your software will only function after a Product Key has been entered by the user. At run-time, the Sentinel License Manager checks that the software is on the machine on which it is licensed to run and that it is being used in accordance with the user's license terms.



Sentinel LDK provides a robust software-based solution using Sentinel SL keys. A Sentinel SL key resides in the secure storage of a specific computer and is patterned on the functionality of a Sentinel HL key.

For more information about Sentinel SL keys, see ["Sentinel SL Keys" on page 28](#).

## Comparative Benefits of Hardware-based and Software-based Solutions

Strong protection and licensing security can be provided with either hardware- or software-based solutions. While many protection and licensing features are common to both options, each also offers specific strengths that might be comparatively limited in the other.

The following table highlights and compares some of the available benefits of hardware- and software-based solutions, and the relative strengths of each option.

Feature	Hardware-based	Software-based
Software and Intellectual Property protection	* * * *	* * *
Secure Licensing	* * * *	* *
Trialware	* *	* * * *
Portability	* * * *	*
Electronic Software Distribution	* *	* * * *
Multiple Feature/Module Licensing	* * *	* * * *

## Advantages of a Combined Solution

As shown in the preceding section, both solutions have their relative strengths in protecting and licensing your software.

You probably utilize various strategies for marketing, selling, and distributing your software. For example, these strategies may include:

- > Determining the level of protection according to the price of the software
- > Determining the level of protection according to market segments, including vertical markets

Your strategies will likely also require the following:

- > The ability to turn trialware into a fully-licensed version using hardware-based or software-based activation
- > The ability to sell software over the Internet, protected with a hardware-based or software-based key

## Sentinel LDK Combined Solution

Sentinel LDK provides the industry's first software DRM solution that combines hardware-based and software-based protection and licensing.

This innovative, self-contained, flexible system enables you to:

- > Implement multiple protection solutions.
- > Define multiple licensing business models according to the requirements of your market, and apply their usage terms independently of the protection process.
- > Select hardware-based (Sentinel HL) or software-based (Sentinel SL) protection keys independently of the protection process.

## Fundamentals of Licensing

---

In addition to protecting your software and intellectual property, you need to protect the revenue from sales of your product. You want to ensure that your software is only available to the appropriate users, according to the terms that you define. This process is controlled by *licensing*.

Licensing provides you with the flexibility to implement your business strategies for your software distribution. When you define the licensing terms on which your software is distributed or sold, you select the terms that are commercially beneficial to your company.

For example, you may decide that you initially want to distribute your software free of charge, so that users can try it before purchasing. You will want to ensure that users can use it for only a limited time before it must be purchased.

Alternatively, you may publish very complex, expensive software. You may decide to make specific components of that software available for a lower price, thus making parts of it accessible to users who cannot afford the full-featured version. Such a decision creates an additional revenue source.

To obtain the maximum benefit from your company's licensing strategy, you need a software licensing system that provides you with the flexibility to tailor licensing terms to fit your business strategies, and to adapt quickly to changes in the market and in your business needs. Your licensing system must also be able to track your defined usage terms along with secure licensing methods.

## Flexible and Secure Licensing Solutions

---

Sentinel LDK gives you the flexibility to choose and apply licensing models and license terms for your protected software on-the-fly. This enables your company to offer attractive software packages and to adapt rapidly to changes in customer purchasing preferences.

## Licensing Planning and Models

An important step in the development of a licensing strategy is the preparation of a licensing plan. Business decision-makers in an organization, such as product or marketing managers, define protection and business rules, and specify the licensing business models required to meet the company's software distribution needs.

A licensing business model is the logic behind a business transaction relating to licensing. For example, a rental licensing business model enables you to charge for the use of software for a specific period of time.

Sentinel LDK enables you to choose from a variety of built-in licensing models, and to customize and build licensing business models and software usage terms to meet your company's individual requirements.

Sentinel LDK supports numerous out-of-the-box licensing business models, that can be used individually or in combination, including:

- > Trialware ("Try before you buy")
- > Rental/Subscription
- > Module-based or Feature-based
- > Floating Usage
- > Time-based
- > Execution-based

You can easily define custom licensing business models and usage terms using the functionality provided by Sentinel LDK. For example, Sentinel LDK functionality enables you to utilize secure read-only and read/write memory storage, flexible counters, and a real-time clock or virtual clock incorporated in the Sentinel protection key.

The separation of the engineering and licensing processes embodied in Sentinel LDK makes it possible to modify your company's licensing strategy as necessary when circumstances change, and to implement these changes quickly and efficiently.

## Updating and Enforcing Usage Terms

When implementing a licensing plan, it is essential to ensure that the software usage terms defined in the plan are securely applied and that licenses reach their legitimate owners. New licenses, and changes and extensions to licenses that have already been deployed, can be subject to tampering if not adequately protected.

Sentinel LDK applies optimal security to the enforcement of usage terms and license extensions. License extensions sent to end users are highly protected, and require the return of a secure receipt. In addition, state-of-the-art Sentinel LDK technology prevents tampering with usage terms.

## Principles of Sentinel LDK

The strength, uniqueness, and flexibility of Sentinel LDK are based on two primary principles:

- > *Protect Once—Deliver Many—Evolve Often*: The concept of separating the Sentinel LDK engineering and business processes.
- > *Cross-Locking*: The technology that supports the *Protect Once—Deliver Many—Evolve Often* concept, enabling a protected application to work with either a Sentinel HL key or a Sentinel SL key.

**NOTE** In general, references in this guide to protected applications are also applicable to protected data files.

### Protect Once—Deliver Many—Evolve Often

At the heart of Sentinel LDK lies the *Protect Once—Deliver Many—Evolve Often* concept. This concept is the process of protecting your software completely independently of the process of defining sales and licensing models.

### Separation of Protection and Business Functions

The engineering process—that is, the protection of your software—is performed by your software engineers using Sentinel LDK Envelope, Sentinel LDK ToolBox and the Sentinel Licensing API protection tools.

The business processes—that is, software licensing and selection of the appropriate Sentinel protection key—are performed by business management using Sentinel LDK-EMS.

As part of the business processes, the *Evolve Often* stage delivers the capability for you and your end users to:

- > Actively track delivery and activation status of end-user entitlements.
- > Track when, how, and by whom your software is being consumed.
- > Easily manage terms of each entitlement using Sentinel LDK-EMS.

The protection processes and the licensing processes—including selection of the appropriate Sentinel protection key type—are performed completely independently of each other.

### Cross-locking

Cross-locking is the Sentinel LDK process that enables you to choose the device to which your protected application and license will be locked—either to a Sentinel HL key or, via a Sentinel SL key, to a specific computer.

The decision about the type of Sentinel protection key to which your software is locked is determined after protection has been implemented—you choose the options that best suit your current business strategies.

## Mixing and Matching Licenses and Sentinel Protection Keys

Sentinel LDK gives you complete flexibility to choose the combination of license and Sentinel protection key that best suits your business requirements. This means that you decide how to bundle your protection, licensing and distribution requirements.

You may choose to release protected software as a downloadable product with a trialware license that, after purchase, is activated with a Sentinel SL key. Additionally, you may choose to ship the same protected software with a network license that is locked to a Sentinel HL key, and allow users unlimited access to all features.

Sentinel LDK offers you an unprecedented number of possible options to combine licenses and Sentinel protection keys.

## Customizing Your Unique Solution

---

Sentinel LDK provides you with a variety of applications and personalized devices that enable you to customize a protection and licensing solution that is appropriate to your business needs:

- > *Sentinel LDK Envelope* enables you to wrap your software in a protective shield at the touch of a button—without having to adjust your source code. It establishes a link between your protected software and a Sentinel protection key, even though the selection of key is determined at a later time.
- > Sentinel LDK Data Protection utility enables you to encrypt data files so that they can only be accessed by specific protected applications. You can additionally apply licensing protection so that the data files can only be accessed when an appropriate Sentinel protection key is present.
- > *Sentinel LDK ToolBox* and the *Sentinel Licensing API* enable you to enhance the protection offered by Sentinel LDK Envelope, by incorporating complex protection mechanisms into your source code.
- > *Sentinel LDK-EMS* enables you to create licenses and lock them to Sentinel protection keys, to write specific data to the memory of a Sentinel protection key, and to update licenses already deployed in the field. These processes are performed independently of the protection process.
- > Custom Developer keys are used in-house by your staff, together with Sentinel LDK state-of-the-art security applications.
- > A selection of *Sentinel protection keys* enable you to meet the specific requirements of your business. Your unique Sentinel protection keys ensure that your applications will only function when the correct key, supplied by you, is present.
- > Additional applications and utilities provide advanced support for these key elements of Sentinel Vendor Suite.

## Personalized Vendor and Batch Codes

When you purchase a Sentinel LDK Starter Kit from Thales, you are provided with Developer keys that contain unique Vendor Codes that are specific to your company. The Vendor Codes are used by Sentinel LDK to communicate with your Sentinel protection keys, and to differentiate your keys from those of other software

vendors.

## Vendor Code

The Vendor Code is a unique code that is assigned to you by Thales when you place your first order for Sentinel protection keys. It is integrated into your Developer keys. At the time you introduce one of your Developer keys to Sentinel LDK, the Vendor Code is extracted from the Vendor key and copied to your machine. When you protect an application using Sentinel LDK Vendor Tools on your machine, the Vendor Code is included in the protected application. This enables the application to communicate with your customers' Sentinel protection key.

## Batch Code

A Batch Code consists of five characters that represent your company's unique Vendor Code. When you order Sentinel protection keys from Thales, you specify your Batch Code, which is then written to the keys before dispatch. To easily identify the Batch Code to which a Sentinel HL key belongs, the Batch Code is written on the outside of each key.

## Selecting the Best Key for Your Requirements

Sentinel LDK protection and licensing are key-based. Your software is distributed with unique actual and/or virtual Sentinel protection keys that you code according to your requirements.

There is a strong inherent link between a protected application and its corresponding Sentinel protection key. Protection is based on making access to the protected application dependent on the presence of a correct Sentinel protection key.

Similarly, when licensing is implemented using Sentinel LDK, the operation of your software is dependent on the presence of a valid license in a Sentinel protection key.

A variety of hardware-based and software-based Sentinel protection keys are available to provide you with the flexibility to sell your software in the ways that are most beneficial to your business goals.

## Sentinel LDK Vendor Keys

When you purchase Sentinel LDK, your Starter Kit contains two Vendor keys—the Master key and the Developer key. These keys enable you to apply protection to your applications, program the Sentinel protection keys that you send to your end users, and to specify the license terms under which your software can be used.

### > Developer key

The Developer key is used by your developers in conjunction with the Sentinel LDK protection tools to protect your software and data files. This key is typically connected to the machine on which Sentinel LDK Envelope executes.

## > Master key

This key is required for Sentinel LDK-EMS only when installed on-premises. The Master key contains your Sentinel LDK Master license and is used by your production staff to create and update licenses and to write data to the memory of a Sentinel protection key. This key is typically connected to the machine on which Sentinel LDK-EMS is installed or where the program that calls Sentinel License Generation API is running.

For Sentinel LDK-EMS hosted by Thales, this key is not required. Store the Master key in a secure location to prevent misuse.

Before you start working with Sentinel LDK, you must introduce one of your Vendor keys to Sentinel LDK. For more information, see the [Sentinel LDK Installation Guide](#).

For more information regarding the Vendor keys and your Sentinel LDK Master license, see "[Understanding the Sentinel LDK Master License](#)" on page 305.

## End-User Keys

Two types of Sentinel protection keys are available:

- > The Sentinel HL key is a physical USB or ExpressCard key that connects to a computer, or a chip that is embedded in the computer.
- > The Sentinel SL key is a software-based key that locks your software to a specific machine. Your software and the user license are locked to the Sentinel protection key that you select.

## Sentinel HL Keys

All Sentinel HL keys—with the exception of Sentinel HL Basic keys—contain internal read/write memory. You can use the memory to do any of the following:

- > Control access to specific software modules and/or packages
- > Assign a unique code to each software user
- > Store licenses from your own licensing schemes
- > Save passwords, program code, program variables, and other data

Sentinel HL keys are distributed with your software to end users. The keys connect to the end users' computers. A variety of Sentinel HL keys are available to suit your requirements. Sentinel HL keys are available in either of two configurations:

- > Sentinel HL (HASP configuration) keys: These keys are fully compatible with software that requires the older HASP HL keys.

**NOTE** Sentinel HL (HASP configuration) keys can be upgraded in the field to Sentinel HL (Driverless configuration) keys. For more information, see "[Upgrading Sentinel HL Keys](#)" on page 350.

- > Sentinel HL (Driverless configuration) keys: These keys provide several advantages over Sentinel HL (HASP configuration) keys:
- (On a Windows machine) Employ HID drivers instead of HASP key drivers. (HID drivers are an integral part of the Windows operating system.) In many cases, it is possible to use these keys without installing any additional support software.
  - (On a Windows machine) Support the use of "AppOnChip" functionality. With AppOnChip, code fragments of selected functions in the protected application are protected from an attacker's eyes by storing them as encrypted data. The code fragments are only decrypted and executed inside the HL key. This provides significantly enhanced security for the application.
  - (On a Linux machine) In many cases, it is possible to use these keys without installing any additional support software.
  - Support a higher number of Features.
  - Provide larger on-key memory space.
  - All Driverless keys (except for Basic keys) support a virtual clock for time-based licenses.
  - All Driverless keys (except for Basic keys) support concurrency (network-based licenses).

Sentinel HL keys offer the highest level of security. In order for a user to access your software, and for it to function correctly, the key must be accessible by the application. Furthermore, Sentinel LDK uses *LicenseOnChip* technology to protect Sentinel HL keys against license tampering.

Sentinel HL keys also have the advantage of portability. This means that the key can be moved from one computer to another. Software may therefore be installed on multiple computers but will only run if the key is connected and authenticated by the software.

**NOTE** A Sentinel HL key can be accessed using a virtual connection. For more information, see ["Virtual Connection of HL Keys" on page 29](#).

**NOTE** Sentinel LDK continues to support the older HASP HL keys. All references to Sentinel HL keys in this document and other Sentinel LDK documents can be understood to include HASP HL keys unless the context of the reference clearly states otherwise.

## Benefits of Sentinel HL Key Protection

Sentinel HL key protection provides the strongest level of protection against piracy. The correct functionality of the software depends on the internal logic of the Sentinel HL key, which is virtually tamper-proof.

In addition, Sentinel HL key protection:

- > Offers the strongest enforcement for license terms, which are stored and protected inside the Sentinel HL key.
- > Enables portability—the software can be used on any computer to which the Sentinel HL key is connected.



- > Does not require transaction with the software vendor to enable activation of the Product.

## Sentinel SL Keys

Sentinel SL keys are virtual, software-based keys that reside in the secure storage of a specific computer. Sentinel SL keys provide the same functionality as Sentinel HL keys, without requiring physical distribution.

Sentinel SL keys are patterned on the functionality of Sentinel HL keys. However, the data is located in the secure storage of the computer on which the Sentinel SL key resides.

After your software is installed on a computer, the end user typically enters a Product Key that is sent, via the Internet or by file transfer, to Sentinel LDK-EMS, together with the fingerprint of the machine. Sentinel LDK-EMS confirms that the Product Key has not been used to activate the software on more than the permitted number of machines—as determined by you—then sends back the Sentinel SL key, which is installed on the end user's machine. This process is also used for updating license terms.

The following types of Sentinel SL keys exist:

- > **SL Legacy** - SL keys that were generated with versions of Sentinel HASP prior to Sentinel LDK v.6.0
- > **SL AdminMode** - SL keys that provide the highest level of security and functionality
- > **SL UserMode** - SL keys that provide a greater level of flexibility under certain circumstances

## Benefits of Sentinel SL Key Protection

With Sentinel SL key protection:

- > Product key activation:
  - Product activation is instantaneous. End users can immediately start using the software with its fully-licensed functionality.
  - The activation process for end users is convenient and transparent.
  - The online connection with end users can enable user registration data to be collected and used for marketing purposes.
- > When using a network license that is locked to a Sentinel SL key, you can specify that a license can be detached from the pool of network seats and attached to a remote recipient machine.

## Sentinel SL Unlocked Licenses

An unlocked license is one that is not locked to a specific machine. An application with an unlocked license (referred to as an *Unlocked Product*) is protected against disassembly. However, the protected application can be duplicated, installed, and used on any machine for as long as the unlocked license allows. Unlocked licenses are used in the following situations:

- > **Trialware products**

The ability to create and distribute trialware products without exposing the protected software to piracy provides a significant marketing advantage when selling software applications. Potential customers can work with the actual application and experience what the application has to offer and how it can benefit the individual or the organization. In addition, anybody that has access to trialware can copy it and distribute it to other people; this multiplies the exposure of the application within the marketplace. Each person who installs and works with the application must, at the end of the grace period (typically 30 to 90 days or 30 executions), decide to purchase an HL or SL key for the application or else be blocked from using the application.

### > Unlocked products

Unlocked products are used when vendors want to protect their applications against reverse engineering but either:

- Have no need to license the application (for example, software that is part of a larger hardware package). The vendor may not need to protect against duplication of the software. However, they want to protect the software against theft of intellectual property.
- Are using a separate product or system to handle licensing of the software.

An unlocked product typically has no time restriction or has a long-term license.

## End-User Keys

Two types of Sentinel protection keys are available:

- > The Sentinel HL key is a physical USB or ExpressCard key that connects to a computer, or a chip that is embedded in the computer.
- > The Sentinel SL key is a software-based key that locks your software to a specific machine. Your software and the user license are locked to the Sentinel protection key that you select.

## Sentinel HL Keys

All Sentinel HL keys—with the exception of Sentinel HL Basic keys—contain internal read/write memory. You can use the memory to do any of the following:

- > Control access to specific software modules and/or packages
- > Assign a unique code to each software user
- > Store licenses from your own licensing schemes
- > Save passwords, program code, program variables, and other data

Sentinel HL keys are distributed with your software to end users. The keys connect to the end users' computers. A variety of Sentinel HL keys are available to suit your requirements. Sentinel HL keys are available in either of two configurations:

- > Sentinel HL (HASP configuration) keys: These keys are fully compatible with software that requires the older HASP HL keys.

**NOTE** Sentinel HL (HASP configuration) keys can be upgraded in the field to Sentinel HL (Driverless configuration) keys. For more information, see ["Upgrading Sentinel HL Keys" on page 350](#).

- > Sentinel HL (Driverless configuration) keys: These keys provide several advantages over Sentinel HL (HASP configuration) keys:
- (On a Windows machine) Employ HID drivers instead of HASP key drivers. (HID drivers are an integral part of the Windows operating system.) In many cases, it is possible to use these keys without installing any additional support software.
  - (On a Windows machine) Support the use of "AppOnChip" functionality. With AppOnChip, code fragments of selected functions in the protected application are protected from an attacker's eyes by storing them as encrypted data. The code fragments are only decrypted and executed inside the HL key. This provides significantly enhanced security for the application.
  - (On a Linux machine) In many cases, it is possible to use these keys without installing any additional support software.
  - Support a higher number of Features.
  - Provide larger on-key memory space.
  - All Driverless keys (except for Basic keys) support a virtual clock for time-based licenses.
  - All Driverless keys (except for Basic keys) support concurrency (network-based licenses).

Sentinel HL keys offer the highest level of security. In order for a user to access your software, and for it to function correctly, the key must be accessible by the application. Furthermore, Sentinel LDK uses *LicenseOnChip* technology to protect Sentinel HL keys against license tampering.

Sentinel HL keys also have the advantage of portability. This means that the key can be moved from one computer to another. Software may therefore be installed on multiple computers but will only run if the key is connected and authenticated by the software.

**NOTE** A Sentinel HL key can be accessed using a virtual connection. For more information, see ["Virtual Connection of HL Keys" on page 29](#).

**NOTE** Sentinel LDK continues to support the older HASP HL keys. All references to Sentinel HL keys in this document and other Sentinel LDK documents can be understood to include HASP HL keys unless the context of the reference clearly states otherwise.

## Benefits of Sentinel HL Key Protection

Sentinel HL key protection provides the strongest level of protection against piracy. The correct functionality of the software depends on the internal logic of the Sentinel HL key, which is virtually tamper-proof.

In addition, Sentinel HL key protection:

- > Offers the strongest enforcement for license terms, which are stored and protected inside the Sentinel HL key.
- > Enables portability—the software can be used on any computer to which the Sentinel HL key is connected.
- > Does not require transaction with the software vendor to enable activation of the Product.

## Sentinel SL Keys

---

Sentinel SL keys are virtual, software-based keys that reside in the secure storage of a specific computer. Sentinel SL keys provide the same functionality as Sentinel HL keys, without requiring physical distribution.

Sentinel SL keys are patterned on the functionality of Sentinel HL keys. However, the data is located in the secure storage of the computer on which the Sentinel SL key resides.

After your software is installed on a computer, the end user typically enters a Product Key that is sent, via the Internet or by file transfer, to Sentinel LDK-EMS, together with the fingerprint of the machine. Sentinel LDK-EMS confirms that the Product Key has not been used to activate the software on more than the permitted number of machines—as determined by you—then sends back the Sentinel SL key, which is installed on the end user's machine. This process is also used for updating license terms.

The following types of Sentinel SL keys exist:

- > **SL Legacy** - SL keys that were generated with versions of Sentinel HASP prior to Sentinel LDK v.6.0
- > **SL AdminMode** - SL keys that provide the highest level of security and functionality
- > **SL UserMode** - SL keys that provide a greater level of flexibility under certain circumstances

## Benefits of Sentinel SL Key Protection

---

With Sentinel SL key protection:

- > Product key activation:
  - Product activation is instantaneous. End users can immediately start using the software with its fully-licensed functionality.
  - The activation process for end users is convenient and transparent.
  - The online connection with end users can enable user registration data to be collected and used for marketing purposes.
- > When using a network license that is locked to a Sentinel SL key, you can specify that a license can be detached from the pool of network seats and attached to a remote recipient machine.

## Sentinel SL Unlocked Licenses

An unlocked license is one that is not locked to a specific machine. An application with an unlocked license (referred to as an *Unlocked Product*) is protected against disassembly. However, the protected application can be duplicated, installed, and used on any machine for as long as the unlocked license allows. Unlocked licenses are used in the following situations:

### > Trialware products

The ability to create and distribute trialware products without exposing the protected software to piracy provides a significant marketing advantage when selling software applications. Potential customers can work with the actual application and experience what the application has to offer and how it can benefit the individual or the organization. In addition, anybody that has access to trialware can copy it and distribute it to other people; this multiplies the exposure of the application within the marketplace. Each person who installs and works with the application must, at the end of the grace period (typically 30 to 90 days or 30 executions), decide to purchase an HL or SL key for the application or else be blocked from using the application.

### > Unlocked products

Unlocked products are used when vendors want to protect their applications against reverse engineering but either:

- Have no need to license the application (for example, software that is part of a larger hardware package). The vendor may not need to protect against duplication of the software. However, they want to protect the software against theft of intellectual property.
- Are using a separate product or system to handle licensing of the software.

An unlocked product typically has no time restriction or has a long-term license.

## Virtual Connection of HL Keys

A standalone Sentinel HL key (that is, a key that does not support concurrency) must typically be physically connected to the machine where the protected application executes. However, you can connect this type of key using a virtual connection using available third-party solutions. These solutions can be used in cases where a physical key cannot be connected due to the lack of a USB port or inability to physically access the machine. You can connect your HL keys to an over-the-network USB solution and access them from any physical or virtual machine as if they were connected locally.

There are several such solutions, both software-based (that can be installed on any PC with a USB port) and dedicated devices. Among the dedicated devices, Thales recommends utnserver Pro, myUTN-2500, dongleserver Pro, and dongleserver ProMAX by SEH Technology. These devices were tested for Thales Sentinel HL keys, and are backed by partnership between SEH Technology and Thales.

## Protection Key Attributes

The various types of Sentinel protection keys that are available provide different levels of security and flexibility, as described in the table that follows.

Type of Sentinel Protection Key	Level of Security	Supported Operating Systems (Local)	Supports Time-based Licenses	Supports Concurrency and Detachable Licenses
SL AdminMode key (excluding Unlocked Products)	++++ <sup>4</sup>	Windows Mac Linux Intel/ARM	Uses V-Clock	Yes <sup>1</sup>
SL AdminMode key (Unlocked Products)	Windows: ++ <sup>4</sup>  Others: + <sup>4</sup>	Windows Mac Linux Intel/ARM	Uses V-Clock	No
SL UserMode key (excluding Unlocked Products)	++++ <sup>4</sup>	Windows Android Linux Intel/ARM	Uses V-Clock	No
SL UserMode (Unlocked Product)	+ <sup>4</sup>	Windows Linux Intel/ARM	Uses V-Clock	No
SL Legacy key	++++ <sup>4</sup>	Windows Mac Linux Intel	Uses V-Clock	Yes <sup>1</sup>
HL (HASP configuration) Basic key	+++++	Windows Mac Linux Intel/ARM	No	No
HL (Driverless configuration) Basic key	+++++ <sup>5</sup>	Windows Mac Linux Intel/ARM Android	No	No
HL (HASP configuration) Pro key	+++++	Windows Mac Linux Intel/ARM	No	No
HL (Driverless configuration) Pro key	+++++ <sup>5</sup>	Windows Mac Linux Intel/ARM Android	Uses V-Clock (Requires V-Clock module in the Sentinel LDK Master license)	Yes <sup>1 2 3</sup> (Detach not supported)
HL (HASP configuration) key (Max, Drive)	+++++	Windows Mac Linux Intel/ARM	No	No

Type of Sentinel Protection Key	Level of Security	Supported Operating Systems (Local)	Supports Time-based Licenses	Supports Concurrency and Detachable Licenses
HL (Driverless configuration) key (Max, Drive)	+ + + + + <sup>5</sup>	Windows Mac Linux Intel/ARM Android	Uses V-Clock	Yes <sup>1 2 3</sup> (Detach not supported)
HL (HASP configuration) Time key	+ + + + +	Windows Mac Linux Intel/ARM	Uses real-time clock on the key	No
HL (Driverless configuration) Time key	+ + + + + <sup>5</sup>	Windows Mac Linux Intel/ARM Android		Yes <sup>1 2 3</sup> (Detach not supported)
HL (HASP configuration) NetTime key	+ + + + +	Windows Mac Linux Intel/ARM		Yes (Detach not supported)
HL (Driverless configuration) NetTime key	+ + + + + <sup>5</sup>	Windows Mac Linux Intel/ARM		
HL (HASP configuration) Net key	+ + + + +	Windows Mac Linux Intel/ARM	No	Yes (Detach not supported)
HL (Driverless configuration) Net key	+ + + + + <sup>5</sup>	Windows Mac Linux Intel/ARM	Uses V-Clock	

**Legend:**

- 1 - Requires network seats from the Sentinel LDK Master license.
- 2 - Requires network seats from the Sentinel LDK Master license. Requires License Manager 7.3 or later on the machine where the protected application executes. The required version of License Manager is provided in Run-time Environment 6.65 or later.
- 3 - To support concurrency for Linux ARM, Run-time Environment 8.11 or later is required.
- 4 - For SL keys, support for virtual machine and for rehost is optional. Adding support for these should be avoided unless required because they implicitly reduce the security level.
- 5 - Security for all HL (Driverless configuration) keys can be further enhanced for Windows platforms using AppOnChip functionality.

**NOTE** The table above relates to the operating system on the machine to which the protection key is attached. However, for protection keys that support concurrency, the protected application can be located on a different machine. In this case, the protected application can be executing under any operating system listed in the table even when that operating system does not support the protection key.

For example: A protected application running on an Android machine can be licensed by an SL AdminMode protection key on a Linux Intel machine.

For information on V-Clock (the virtual clock available on most Sentinel protection keys), see ["How Sentinel LDK Protects Time-based Licenses With V-Clock" on page 346](#).

For full technical specifications of the available Sentinel HL keys, refer to the [Sentinel HL Data Sheet](#).

For additional information, see ["Situations That Require Sentinel LDK Run-time Environment" on page 194](#).

## Sentinel LDK Protection Process

When you are developing your software, your engineers integrate a variety of calls to data stored in the memory of the Sentinel protection key.

## Encryption and Decryption

Sentinel LDK encryption and decryption are based on the Advanced Encryption Standard (AES) algorithm. The encryption secret of the algorithm is stored in the Sentinel protection key. To enhance security, all communication between an application and a Sentinel protection key is randomly encrypted. This inhibits emulation of a Sentinel protection key.

## Obtaining Additional Information About Sentinel LDK

You can find more information about Sentinel LDK in the help systems for the various Sentinel LDK tools. In addition, much of the Sentinel LDK documentation is available online at [Sentinel LDK documentation](#).

For additional assistance, you can contact our [Customer Support Team](#).



## CHAPTER 2: Understanding Cloud Licensing

This section describes an additional model for generating and distributing software-based licenses, based on an extension to software-based protection keys (described in "[Understanding Sentinel LDK Software Protection and Licensing](#)" on page 15).

### Overview

---

Cloud-based software licensing is a method in which licenses reside in the virtual cloud. Typically, these licenses are subscription licenses, which customers need to renew monthly or yearly (or any other increment that the vendor decides). There are many benefits to a cloud-based subscription license both for the company licensing a product and its customers. Cloud licensing allows companies to constantly update their software and deliver it easily to customers, maintain a constant income stream, and more.

The cloud licensing model provides a simpler mechanism, both for the vendor and for the end user, for distributing and managing licenses.

Using this licensing model, the vendor generates and installs all required product licenses on a single license server machine with Internet access. The vendor generates and distributes a unique set of credentials or a license string for each end user.

The end user logs in to the protected application with the credentials or installs the license string on their machine. The end user can access the license server and consume a license to execute the protected application online or detach a license and then run the protected application offline.

### Implementing Cloud Licensing

---

The software vendor implements cloud licensing using one of the following methods:

#### **Identity string-based licensing**

Identity string-based licensing enables end users to access the vendor's application or service using a unique identity string. This identity string must be installed on the client device where the vendor's application runs. When the vendor's application tries to launch on that device, the client communicates with the cloud licensing service on the vendor's or Thales' service-hosted, cloud license manager server, which authenticates the client identity based on the identity string and verifies the access permissions assigned to the machine account. If the machine account is authorized to access the application, the application opens and runs.

Identity strings are generated and managed using one of the following methods:

- > **Cloud Licensing Portal.** The cloud licensing service is hosted on the vendor's server. For more information, see ["Sentinel LDK Cloud Licensing Portal" on page 157](#).
- > **Admin Control Center.** The cloud licensing service is hosted on the vendor's server. For more information, see ["Cloud Licensing Overview" on page 171](#).

## Benefits of Cloud Licensing

---

Some of the benefits provided by the cloud licensing model are as follows:

- > **License Mobility.** End users can consume licenses from anywhere. Once they receive and install the credentials or identity string, they can execute the application, regardless of where they are.
- > **Implementation.** Cloud licenses are easy for vendors to implement, with no compromises on security. No changes to Sentinel Licensing API are required. The model is fully supported by Sentinel LDK Envelope. Licenses are easy to deploy.
- > **End-user control.** Identities or credentials can be disabled by the vendor at any time, or be limited to a certain number of machines.
- > **VM solution.** Cloud licensing provides a simple licensing solution for virtual machines and containers (such as Docker) that is both secure and fully agnostic. There is no need to deal with fingerprints, no risk of cloning (with the appropriate clone protection scheme), and no risks of snapshot-restore attacks. All that is required is connectivity.
- > **Clone protection without fingerprint issues.** Users can easily upgrade hardware and update the operating system.
- > **Secure license information.** Secure storage (license information) has increased security and reliability, and is inaccessible to end users. As a result, it cannot be deleted or reverted. Since the secure storage is accessible to the vendor, it can be deployed on highly dependable RAID arrays and be backed up regularly.
- > **Business insight.** Since the vendor hosts the server, they can view usage data and patterns that would otherwise not be available.
- > **Manage trials.** The vendor can easily manage trials by creating a single SL license and then providing users with expiring credentials or client identities that consume the same license. For example, the vendor can host a single cloud-enabled SL key with a perpetual license and, as needed, create identity strings that are active for 30 days to be provided to evaluators.
- > **Provide emergency cloud licenses.** The vendor can provide customers with emergency limited-time credentials or client identities in case they face issues with their local HL or SL key.
- > **Manage user access to network seats.** An IT administrator can manage which users can access network seats and for how long by distributing client identities with expiration dates.

Identity string-based licensing supports detaching licenses. Working with detached licenses provides the following additional benefits:

- > Less demanding on server and network infrastructure. Communication to the server is limited to only a single detach operation. This also removes much of the need for server redundancy.
- > Performance identical to local licenses. Once a license is detached, consuming it is unaffected by network latency.
- > No need for constant connectivity. Once the license is detached, the user's machine can remain offline.

**NOTE** While intended primarily for implementation by software vendors, the cloud licensing model can also be implemented by the vendor's customers for distribution of identity strings within their organization.

## Part 2 - Protection

### In this section:

---

- > ["Protecting Software" on page 37](#) – Provides an overview of Sentinel LDK software protection, including its fundamental elements, a summary of how it works, and an introduction to Sentinel LDK protection methods.
- > ["Sentinel Licensing API Protection" on page 43](#) – Provides an overview of the Sentinel Licensing API, details the prerequisites for using the API, introduces the Sentinel LDK ToolBox application and describes the functionality of the API.
- > ["Sentinel LDK Envelope Protection" on page 52](#) – Provides an overview of software protection using Sentinel LDK Envelope, details the prerequisites for using the application, and describes its functionality. In addition, it describes the Sentinel LDK Envelope protection parameters and how to encrypt data files.
- > ["Protection Strategies" on page 72](#) – Outlines strategies for maximizing Sentinel LDK protection, including best practices and optimizing the use of the Sentinel Licensing API and Sentinel LDK Envelope.
- > ["Protecting Data Files" on page 77](#) – Describes data file protection using the Sentinel LDK Data Protection utility. It includes information about the types of protection that are available.

# CHAPTER 3: Protecting Software

This section provides an overview of Sentinel LDK software protection, including its fundamental elements, a summary of how it works, and an introduction to Sentinel LDK protection methods.

*In this section:*

- > ["Sentinel LDK Protection " below](#)
- > ["Elements of Sentinel LDK Protection" on the next page](#)
- > ["Selecting a Protection Method" on page 41](#)

## Sentinel LDK Protection

Sentinel LDK is an innovative, advanced solution for protecting software against illegal or unauthorized use. The solution deters illegal access and execution of protected applications.

A deployed application that is protected with Sentinel LDK requires access to a specific Sentinel protection key in order to run. The protected application queries the Sentinel protection key for predefined information. If the Sentinel protection key is not present, or the information returned is incorrect, the program does not execute, or stops functioning.

After you have selected a Sentinel LDK protection method, implementation is straightforward. Regardless of the selected protection strategy, protected applications only work correctly if they can access the information stored in a specific Sentinel protection key.

The following table provides an overview of how the tools available for protection and licensing can be used to achieve the desired results:

Objective	Sentinel Licensing API	Sentinel LDK Envelope	Combined Implementation (Licensing API and Envelope)
Copy protection	Provides basic protection.	<b>Provides very strong protection.</b>	<b>Provides very strong protection.</b>
Anti-reverse engineering	Not included	<b>Included</b>	<b>Included</b>

Objective	Sentinel Licensing API	Sentinel LDK Envelope	Combined Implementation (Licensing API and Envelope)
Number of Features that can be licensed in a given application	<b>Unlimited. Each Feature is integrated separately using a unique Feature ID and login API call.</b>	One Feature per executable or DLL. No API integration is required.	<b>Unlimited. Each Feature is integrated separately using a unique Feature ID and login API call.</b>
Support for custom licensing	<b>Available using protection key memory.</b>	Not supported.	<b>Available using protection key memory.</b>

**NOTE** For the most flexible licensing and highest level of copy protection, Thales recommends that you implement both Sentinel Licensing API and Sentinel LDK Envelope.

## Elements of Sentinel LDK Protection

The Sentinel LDK protection system is based on the following:

- > Protecting programs and data files
- > Identifying the Sentinel protection key
- > AES encryption
- > Confidential protection parameters
- > Utilizing Protection Key memory
- > Anti-debugging and reverse engineering measures

### Protecting Programs and Data Files

Sentinel LDK provides two primary protection methods:

- > Sentinel LDK Envelope
- > Sentinel Licensing API

When you protect your software using either of these methods, you are essentially forming an inherent link between the protected application and a specific Sentinel protection key.

## What Can Be Protected

Sentinel LDK enables you to protect a variety of applications and data files. You can apply protection directly to:

- > Compiled executables, DLLs and .NET assemblies
- > Specific functions or entire programs. Sentinel LDK protects all levels of software from function level to entire programs
- > Sensitive data and intellectual property

All the above are protected against any attempt at reverse engineering.

For additional information about the available protection parameter options, see the following sections:

- > ["Sentinel Licensing API Protection" on page 43](#)
- > ["Protecting Data Files" on page 77](#)

## Availability of the Sentinel Protection Key

The Sentinel protection key, or to be more precise—the intelligence contained within the Sentinel protection key—is the primary component of the Sentinel LDK protection system.

The main factor governing Sentinel LDK protection is whether a deployed program can identify and access the intelligence contained in a specific Sentinel protection key at run-time. This factor is unambiguous—*the Sentinel protection key is either available or is not available!*

Regardless of the protection method adopted, protected applications only function when they can access the required information contained in a specific Sentinel protection key.

Sentinel protection keys, and their ‘intelligence’ cannot be cloned to replicate the link between them and the protected application.

## AES Encryption

A protected application relies on the ‘intelligence’ in the memory of a specific Sentinel protection key in order to function. In addition to the checks for the Sentinel protection key, data can be encrypted and decrypted using the intelligence available in the Sentinel protection key.

## AES Encryption and Decryption

The encryption engine in the Sentinel protection key is based on the AES algorithm. Sentinel LDK encryption uses a set of confidential 128-bit encryption keys that remain in the Sentinel protection key.

Your protection schemes should always involve greater sophistication than merely confirming the presence of the required Sentinel protection key. However, verifying the required Sentinel protection key through data encryption and decryption requires forward planning. First, encrypted data must be available. This data must then be sent to the Sentinel protection key, where it is decrypted.

If the data is correct, the Sentinel protection key is considered to be “present.” For additional information, see ["Time Functions" on page 51](#).

## Confidential Protection Parameters

The essence of software protection is confidentiality. Without confidential elements, any software security system is vulnerable to attack.

## Vendor Code

Each software vendor who uses Sentinel LDK is assigned a unique *Vendor Code*.

The Vendor Code forms an integral part of the protection parameters that constitute the link between the protected applications and the Sentinel protection key. However, the Vendor Code is only part of the link. The code merely provides the protected software with access to the Sentinel protection key and its resources. The Vendor Code is required in order to call Encrypt and Decrypt API functions, call memory read/write API functions, and consume licenses.

Access to the Vendor Code does not allow an attacker to create licenses, remove Envelope protection, or perform activities that would typically be regarded as license abuse. Therefore, while the Vendor Code should be kept confidential, the code on its own is not sufficient to enable unauthorized use of the protected software.

All Sentinel LDK protection applications require the Vendor Code. For information on how to access the code, see ["Extracting the Vendor Code from Sentinel Vendor Keys" on page 45](#).

## Utilizing Protection Key Memory

The secure memory on Sentinel protection keys can be utilized (read and write) as a component of the protection scheme for the software. Confidential data can be stored in the Protection Key memory, including snippets of program code, the customer name, or any other data.

Use the memory editors included in Sentinel LDK ToolBox to read or write data in the Protection Key memory. For additional information, see ["Memory Functions" on page 51](#).

In your production environment, use Sentinel LDK-EMS or Sentinel License Generation API to handle Protection Key memory.

## Anti-Debugging and Reverse Engineering Measures

Sentinel LDK protects intellectual property and provides the functionality to combat anti-debugging and reverse engineering. Anti-debugging and reverse engineering usually try to unravel the protection scheme of protected software by tracing a compiled application to its source code. Sentinel LDK Envelope implements contingency measures to ward off such attacks and prevent crackers from uncovering algorithms used inside protected software.



## Selecting a Protection Method

Sentinel LDK offers two software protection methods; *Sentinel Licensing API* and *Sentinel LDK Envelope*. Both methods establish an inherent link between the protected software and the intelligence contained in a specific Sentinel protection key.

When selecting a protection method, the following issues must be considered:

- > What the Sentinel protection key should protect
- > How the Sentinel LDK protection parameters are best applied
- > Whether the time required to implement the solution is a critical factor
- > Whether flexibility in implementing the protection scheme is important

These issues are discussed in the following sections.

### What to Protect

When protecting software with Sentinel LDK, there are various options for applying protection. Sentinel Licensing API is used to protect the software before it is compiled. Protection can also be applied after the software is compiled using Sentinel LDK Envelope. You can choose whether to apply protection to an entire program, a subprogram, or simply to a Feature.

You may opt to use either the Sentinel Licensing API or the Sentinel LDK Envelope protection method, or both, depending on your specific requirements. Use the following table to determine which method best meets your specific requirements.

Sentinel LDK Envelope	Sentinel Licensing API
Quick, automatic protection process that shields your software	Manual implementation of calls to Sentinel Licensing API
Define specific protection parameters for your programs	Controlled process ensuring maximum security. The strength of protection is proportional to the degree to which the Sentinel Licensing API functionality is invested in implementation.
No source code required	Source code must be available
Anti-debugging and reverse engineering measures provided	Maximum flexibility

## Importance of Control Over the Protection Scheme

When applying protection using Sentinel Licensing API, you control the entire protection process. You determine when the protected application queries the Sentinel protection key, and how it should behave in different scenarios. With Sentinel LDK Envelope, compiled programs are wrapped with random protection parameters. If you run Sentinel LDK Envelope twice to protect the same program, two different output files are produced with different protective modules and shields.

## Significance of the Time Factor

When a high protection level is specified in Sentinel LDK Envelope, file size increases and the protected application takes longer to launch. Consider this factor when you are deciding on the protection level settings that you choose. Aim for the optimal balance between protection level and launch time.

## How to Apply Protection

When using the Sentinel Licensing API, protection is integrated at the source code level in a carefully considered manner. You determine where in the source code to place calls to the Sentinel Licensing API.

Sentinel LDK Envelope offers an automated, speedier method of protecting software. You define settings for protection parameters that are applied to protected applications.

**NOTE** When enabling or disabling some features you might reduce the level of protection provided by the software.

# CHAPTER 4: Sentinel Licensing API Protection

This section describes the Sentinel Licensing API protection method.

*In this section:*

- > ["Sentinel Licensing API Overview" below](#)
- > ["Sentinel Licensing API Prerequisites" on the next page](#)
- > ["Learning About the Sentinel Licensing API" on page 46](#)
- > ["Sentinel Licensing API Implementation" on page 47](#)
- > ["Sentinel Licensing API Functionality" on page 50](#)

**NOTE** The Sentinel Licensing API is not applicable for protecting data files.

## Sentinel Licensing API Overview

The Sentinel Licensing API is a robust method of software protection, the strength of which is wholly dependent on its implementation.

The extent to which the functionality afforded by the Sentinel Licensing API is utilized, determines the overall level of software security. To fully utilize the protection offered by the Sentinel Licensing API, strive to maximize the complexity and sophistication of your implementation.

It is essential that, before protecting your application, you are familiar with the overall functionality of the Sentinel Licensing API. For a description of the functions that make up the Sentinel Licensing API, see the Sentinel Licensing API Reference for the relevant code language:

[C](#) | [.NET](#) | [.NET Standard](#) | [Java](#) | [Web Service \(REST API\)](#)

To protect your software using the Sentinel Licensing API, you insert calls to a Sentinel protection key throughout your application's source code. You can add calls to your application that check for the presence of a Sentinel protection key at any point during run-time, and you can designate responses to these checks. For example, if the required Sentinel protection key is not found, you might specify that the protected application suspend or terminate itself.

Your application can also check the memory of a Sentinel protection key for specific data. In addition, you can use the Sentinel Licensing API to encrypt or decrypt data.

To facilitate a speedy learning curve, Thales recommends that you familiarize yourself with and test specific Sentinel Licensing API functions using Sentinel LDK ToolBox. Sentinel LDK ToolBox is a GUI-based application that interfaces with various Sentinel LDK APIs. For additional information, see ["Learning About the Sentinel Licensing API" on page 46](#).

Sentinel LDK also includes Sentinel Licensing API sample folders for specific compilers. Each Sentinel LDK interface includes a sample application demonstrating API usage and a specific header file. The sample applications are located in the **Samples** in the Sentinel LDK installation on the Windows, Linux, or Mac machine.

## Universal Sentinel Licensing API

The Sentinel Licensing API is a universal API that works with all Sentinel protection keys. Sentinel Licensing API implementation and usage is independent of the Sentinel protection key you use.

Utilization of the Sentinel Licensing API is independent of the access mode used to search for a specific Sentinel protection key. The same Sentinel Licensing API functions are used to enable programs' access to remote Sentinel protection keys, or Sentinel protection keys that are present locally.

## Sentinel Licensing API Prerequisites

You may have to install the Sentinel LDK Run-time Environment in order to enable the Licensing API. For more information, see ["Situations That Require Sentinel LDK Run-time Environment" on page 194](#).

## Vendor Code

It is necessary to provide the Vendor Code in order to access a Sentinel protection key and its resources, including memory. Vendor Codes are usually stored in the **VendorCodes** directory. The location of the directory is described later in this topic.

In the Sentinel LDK Demo Kit, customers are provided with Sentinel HL Demo keys that work with the DEMOMA Vendor Code. This Vendor Code can be used to apply protection with the Sentinel Licensing API.

**NOTE** Do not distribute software protected with a Sentinel HL Demo key. This Sentinel protection key is only for evaluation purposes.

The Starter Kit you receive for Sentinel LDK contains two Vendor keys—a Developer key and a Master key. These keys contain your company's unique Vendor Code. The Developer key is used by engineers for adding protection to your software. With Sentinel LDK-EMS hosted by Thales, you do not require the Master key; however, you should store it in a secure location to protect against misuse. With Sentinel LDK-EMS installed on premises, the Master key is used for producing licenses and orders, and it must be connected to the machine where Sentinel LDK-EMS is installed.

Sentinel Vendor Suite applications (Sentinel LDK Envelope, Sentinel LDK ToolBox, and Sentinel LDK-EMS) must recognize and have access to the unique Vendor Code that was assigned to you when you received your Starter Kit. Your Vendor Code is stored inside each Vendor key. You introduce one of the Vendor keys using the Master Wizard, as described in the following section.

For more information, see ["Vendor Code" on page 40](#).

## Extracting the Vendor Code from Sentinel Vendor Keys

You need to extract the Vendor Code from your Vendor keys so that the Sentinel LDK system will recognize it when you are working with any of the Vendor Suite applications. The Master Wizard extracts the Vendor Code for you.

Depending on your Sentinel LDK configuration, if you launch a Sentinel Vendor Suite application, and you have connected a new Vendor key to your computer, the Master Wizard will launch automatically. Alternatively, you can launch the Master Wizard manually.

For detailed information on using the Master Wizard, see the section on introducing Vendor keys in the [Sentinel LDK Installation Guide](#).

By default, your Vendor Code information is saved in the following directory:

```
%UserProfile%\Documents\Thales\Sentinel LDK 10.0\VendorCodes
```

## Vendor-specific File Naming Conventions

The format of a Vendor Code file name is *BatchCode.hvc*. For example, if your Batch Code is **W3FLY**, the file name will be **W3FLY.hvc**. (The Batch Code is a representation of your Vendor Code.) Your Vendor keys and all your Sentinel HL keys are labeled with your Batch Code.

By default, Sentinel Vendor Suite applications search the **VendorCodes** folder for your Vendor Code/Batch Code information.

## Licensing API

At the time the Master Wizard extracts the Vendor Codes, it downloads your customized Licensing API libraries from Thales servers.

Your customized API libraries are saved in the following directory:

```
%UserProfile%\Documents\Thales\Sentinel LDK 10.0\API\Licensing
```

The APIs for each language are stored in a separate subdirectory. The format of API library names (for Windows) is *hasp\_windows\_vendorID.libraryExtension*.

## Example

For C applications, the following libraries are downloaded:

Library	Description
hasp_windows_vendorID.dll	Dynamic version of the API library

Library	Description
hasp_windows_vendorID.lib	Implemented library for hasp_windows_vendorID.dll library
libhasp_windows_vendorID.lib	Static version of the API library
libhasp_windows_bcc_vendorID.lib	Static version of API library compiled with Borland C compiler

*vendorID* represents the vendor ID for your Batch Code or **demo** for the DEMOMA Batch Code.

## Learning About the Sentinel Licensing API

There are two components of Sentinel LDK that enable you to study how the Sentinel Licensing API works, and its range of capabilities.

- > **Sentinel LDK ToolBox:** A utility with a graphical user interface that is part of Sentinel Vendor Suite. For more information, see ["Sentinel LDK ToolBox" below](#).
- > **Sentinel Licensing API Samples:** A set of examples for implementing the Sentinel Licensing API. For more information, see ["Sentinel Licensing API Samples" on the next page](#).

### Sentinel LDK ToolBox

Sentinel LDK ToolBox is an interactive interface to work with various Sentinel LDK APIs. You can execute calls to the Sentinel Licensing API using Sentinel LDK ToolBox. The calls are then relayed to a Sentinel protection key.

To use Sentinel Licensing API with Sentinel LDK ToolBox you must have a Developer key and a valid Vendor Code so that you can access Sentinel protection keys. Sentinel LDK ToolBox is started from the Sentinel LDK launcher. For more information, see the [Sentinel LDK ToolBox User Guide](#).

**NOTE** Sentinel LDK ToolBox does not support the Sentinel Licensing REST API.

### API-related Functionality

Sentinel LDK ToolBox serves as a training tool for the Sentinel APIs. Sentinel LDK ToolBox functionality enables you to:

- > Display the source code generated for each function call. This generated source code can be copied and pasted into your application source code.
- > Evaluate manual implementation of each API. Every API function included in Sentinel LDK ToolBox is displayed on a separate screen. To execute a function call, you provide specific information related to the selected function.
- > Transfer memory buffers to the AES encryption engine in a Sentinel protection key. The program can also be used to decrypt data buffers.

- > Create multiple programming language interfaces for the various APIs.

## Sentinel Licensing API Samples

Sample applications are provided to demonstrate how to implement Sentinel Licensing API protection in your source code. The samples demonstrate how the API functions work.

Your Sentinel LDK installation contains folders for various interfaces and compilers. Each folder includes the required API libraries, a header file and a sample application. The Sentinel HL Demo key—marked DEMOMA—must be connected to your computer when using the sample applications.

The Sentinel Licensing API samples are available in the Sentinel LDK directory structure. In Windows, for example, the samples are located under `%ProgramFiles(x86)%\Thales\Sentinel LDK\Samples\Licensing`.

## Sentinel Licensing API Implementation

---

This section describes the pre-implementation issues you should consider, and the workflow for implementing the Sentinel Licensing API. It also provides an overview of how to log in to and out of a session.

### Planning Your Requirements

Before implementing the Sentinel Licensing API, the following preliminary issues should be considered.

#### > What do you want to protect?

This may seem obvious, but it is crucial when you decide where to place the calls to the Sentinel protection key. Typically, you would want to verify the presence of the Sentinel protection key at startup. However, you can also identify certain aspects of the software to protect, and apply your Sentinel Licensing API calls accordingly.

#### > Will encrypted data be included in your implementation scheme?

If you plan to use encrypted data at run-time, use Sentinel LDK ToolBox to encrypt the data. Insert the encrypted data when implementing the Sentinel Licensing API. The data is decrypted at run-time by the Sentinel protection key.

#### > Is data going to be stored in the Protection Key?

If the software is protected by a Sentinel protection key with memory functionality, sensitive data can be stored in the Sentinel protection key. The Sentinel Licensing API enables access to read from or write to Protection Key memory. Use Sentinel LDK ToolBox to write data buffers to Protection Key memory.

## Sentinel Licensing API Workflow

After planning what data is going to be protected and how that protection will be applied, you are ready to protect your application with the Sentinel Licensing API.

The recommended workflow for implementing the Sentinel Licensing API is as follows:

1. Study the code of the sample application corresponding to your development environment.
2. In your application source code, insert a login call to the Sentinel protection key. A successful login establishes a login session. The login session has its own unique handle identifier.

**NOTE** The session identifier is self-generated and applies to a single login session. For more information, see the description of the **LoginScope** function in the [Sentinel Licensing API Reference](#).

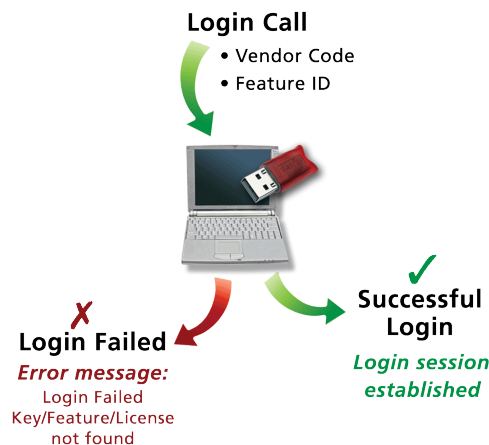
3. After a login session is established, you can use other Sentinel Licensing API functions to communicate with the Sentinel protection key. For example, you can use the **Decrypt** function to decrypt important data used by your application. You can also read data stored in the Protection Key memory, set timestamps, and other actions.
4. Using the output generated in Step 3, check for potential mismatches and notify the user accordingly.
5. Repeat steps 2–4 throughout the code.
6. Compile the source code.

**NOTE** After you have compiled the source code, use Sentinel LDK Envelope to add an extra layer of protection to your software. This process also prevents reverse engineering of protected code.

## Sentinel Licensing API Login Function

The `login` function is the gateway to Sentinel Licensing API implementation. You must open a successful login session to search for and communicate with a Sentinel protection key. To log into a Sentinel protection key, you need to provide a Feature ID and a valid Vendor Code.

If the Sentinel protection key is not accessible by the computer, an error message is displayed. An error message is also displayed if the declared Vendor Code is not valid for a detected Sentinel protection key.





## Figure 1: Sentinel LDK Login Operation Summary

### Login Options

When using the Sentinel Licensing API implementation, login calls are not dependent on specific Sentinel protection keys. However, when performing login calls you must specify what it is that you are actually logging into. When logging in you must declare:

- > If you are logging into a default or a specific Feature
- > How to search for the Sentinel protection key
- > How the login counter should be handled
- > Whether to enable or disable connection to the Sentinel protection key via a terminal server

### Declaring Feature IDs

You can either log into a specific Feature, or to the default Feature stored in the Sentinel protection key. The default Feature is assigned Feature ID 0.

When logging into a licensed Feature, the protected application not only checks for the presence of the Sentinel protection key, it also checks the terms of the license contained in that key. If the license is valid, the Feature is enabled.

### Controlling Login Calls

Additional aspects of a login call can be controlled when implementing the Sentinel Licensing API, as follows:

- > Search options
- > Login counter
- > Terminal server detection
- > Enabling access to Sentinel HL v.1.x keys

Each aspect is described below.

#### Search Options

The default search setting enables a protected application to search both the local computer and the network for the required Sentinel protection key. You can limit the Sentinel protection key search option, as follows:

- > Search only the local PC for a Sentinel protection key
- > Search only the network for a connected Sentinel protection key

## Login Counter

By default, when a Sentinel LDK license is accessed in a Sentinel HL network key, license usage is determined by counting the number of workstations that use the protected application. You can change this condition so that license usage is based on the number of protected application processes that are in use.

## Access to Legacy Memory on Sentinel HL Key

By default, the Sentinel LDK system does not enable access to the legacy memory on Sentinel HL keys. To override this restriction, select the **Allow access to Sentinel HL v.1.x** check box in the Sentinel LDK ToolBox Settings window.

**NOTE** Every Sentinel protection key login session must be terminated with a corresponding logout call.

# Sentinel Licensing API Functionality

The extent of the protection afforded by the Sentinel Licensing API is dependent on the way that it is implemented. Calls to a Sentinel protection key that are inserted in the source code control access to the application at run-time.

This section describes the Sentinel Licensing API options that are available after a successful login session is established. For a detailed discussion about how to optimize your Sentinel Licensing API implementation, see ["Protection Strategies" on page 72](#). For a demonstration of how the Sentinel Licensing API works, use Sentinel LDK ToolBox.

## Function Groups

Sentinel Licensing API functions are categorized into five groups, based on common functionality and linkage.

- > Session functions
- > Encryption/Decryption functions
- > Memory functions
- > Time functions
- > Management functions

## Session Functions

A session is created by executing a successful login call to a license residing in a specific Sentinel protection key. For more information about logging in, see ["Login Options" on the previous page](#). At the end of a session, use the `logout` function to close the session.

## Encryption Functions

You can encrypt or decrypt data buffers using the AES-based encryption engine in the Sentinel protection key. The encryption engine uses symmetric encryption. This means that the same encryption key is used later to decrypt the data buffer.

## Memory Functions

Use the memory to store data to be used by the application at run-time, and information that can be used later to verify and identify an end user. Control of access to sensitive data forms an integral part of your protection scheme.

The Sentinel Licensing API can be used to:

- > Read data buffers stored in the Protection Key memory
- > Write data buffers to the Protection Key memory

The size of the data buffers is restricted by the memory available in the specific Sentinel protection key type. For information about the memory capacity of the available Sentinel protection keys, refer to the [Sentinel HL Data Sheet](#).

## Time Functions

Sentinel Licensing API can be used to access:

- > the real-time clock in a Sentinel HL Time key or Sentinel HL NetTime key
- > the V-Clock in a Sentinel HL (Driverless configuration) key. For more information, see "[How Sentinel LDK Protects Time-based Licenses With V-Clock](#)" on page 346.

This functionality enables you to read the time. Two date and time conversion functions are included in the Sentinel Licensing API.

## Management Functions

The Sentinel Licensing API includes functions that enable you to retrieve information on the system components, the current login session, the status of a deployed Sentinel protection key, and license updates.

When using Sentinel SL keys, the **Transfer** function enables you to:

- > detach a license from a pool of network seats.
- > rehost a protection key from one of a customer's machines to another.

You can also use the **Update** function to install updates. You do not need to be logged in to a session in order to perform this function. For additional information, see the [Sentinel Licensing API Reference](#).

# CHAPTER 5: Sentinel LDK Envelope Protection

This section describes software protection using Sentinel LDK Envelope.

*In this section:*

- > ["Functionality" below](#)
- > ["Sentinel LDK Envelope for Windows" on page 56](#)
- > ["Protecting .NET Assemblies" on page 62](#)
- > ["Protecting Python Applications" on page 66](#)
- > ["Sentinel LDK Envelope for Linux Applications" on page 67](#)
- > ["Sentinel LDK Envelope for Mac Binaries" on page 67](#)
- > ["Sentinel LDK Envelope for Java Executables" on page 69](#)

## Functionality

Sentinel LDK Envelope is a wrapping application that protects your applications with a secure shield. This application offers advanced protection features to enhance the overall level of security of your software.

Sentinel LDK Envelope protects Win32, Windows x64, and .NET executables and DLLs, and Java executables—providing a means to counteract reverse engineering and other anti-debugging measures.

Sentinel LDK Envelope can also be used to protect Mac executables and dynamic shared libraries (Mach-O) (see ["Sentinel LDK Envelope for Mac Binaries" on page 67](#) for more information) , and Linux executables and shared objects (see ["Sentinel LDK Envelope for Linux Applications" on page 67](#) for more information).

**NOTE** The words *program* and *application* are used throughout this section as a generic reference to the various types of programming code that can be protected using Sentinel LDK Envelope, regardless of whether they are executables, binaries, assemblies, libraries or shared objects.

Sentinel LDK Envelope is not used directly to *protect* data files. However, it can enable a protected application to access and write data to a protected data file.

By using Sentinel LDK Envelope to protect your application, you establish a link between the protected application and a Sentinel protection key. This link is broken whenever the protected application cannot access the required Sentinel protection key.

Implementing Sentinel LDK Envelope protection is the fastest way to secure your application without requiring access to your software source code.

Sentinel LDK Envelope provides both graphical user interface (GUI) and command-line utility options. The graphical interface enables you to:

- > Protect Windows and .NET executables and DLL files, and Java executables
- > Enhance the protection of .NET and Java executables by defining Method-level protection
- > Protect Mac Mach-o binaries
- > Protect 64-bit Linux executables and shared objects
- > Define a variety of global protection parameters for your program
- > Specify a Vendor Code to authenticate the presence of a specific Sentinel protection key
- > Customize the run-time messages that will be displayed to end users running protected applications

In addition to linking protected applications to a specific Sentinel protection key, Sentinel LDK Envelope wraps the application file with numerous protection layers that are randomly assembled.

**NOTE** The random multi-layer wrapping of protected applications by Sentinel LDK Envelope ensures that implemented protection strategies differ from one protected application to another.

Command-line utilities enable you to protect:

- > Win32, Windows x64, and .NET executables and DLL files
- > Java executables
- > 64-bit Linux executables and shared objects
- > Mac binaries

The command-line utilities also enable you to easily apply protection parameters that were defined using the Sentinel LDK Envelope GUI. This simplifies the process of reapplying protection parameters to your application during the development process.

## Basic Protection Workflow

This section provides a workflow that describes the elements of protecting applications using Sentinel LDK Envelope. Additional information about specific procedures is provided in the [Sentinel LDK Envelope for Windows](#).

1. Launch Sentinel LDK Envelope from Sentinel LDK Launcher.
2. Add the executable, library, or .NET assembly you want to protect to the project.

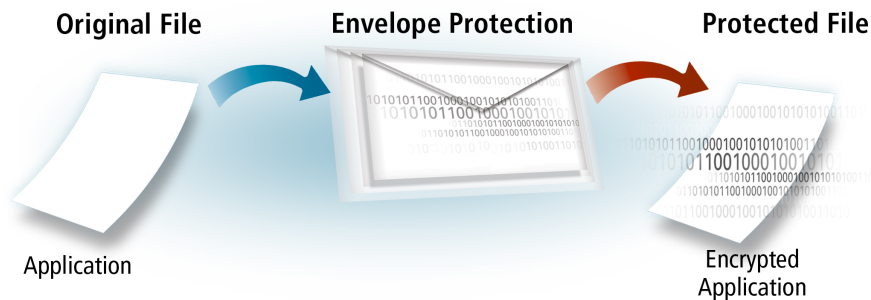
3. Define protection parameters for the protected application.
4. Protect the program.
5. Distribute the protected software together with your encrypted Sentinel protection keys.

**NOTE** Sentinel LDK Envelope does not affect the files being protected. However, it is highly recommended that you designate a separate output folder for the protected application in order to distinguish between source (unprotected) and output (protected) files.

Sentinel LDK Envelope protection involves the application of protection parameters that are controlled by the engines running Sentinel LDK Envelope. You apply these parameters to an unprotected source.

Sentinel LDK Envelope does not affect the original files or the way a protected application actually works. The only modification is that user access is conditional on the presence of a required Sentinel protection key. If the Sentinel protection key is present, the protected file runs.

The logic of Sentinel LDK Envelope protection is illustrated in the following diagram. Note that the original file can be a Win32, or Windows x64 executable or DLL; a Windows .NET assembly executable or dynamic library; a Java executable; a Linux executable or shared object; or a Mac binary.



To ensure the highest level of security for your software, Sentinel LDK Envelope for Win32 removes debugging data from the programs that it is protecting.

It is recommended that Linux software engineers strip extraneous symbols from the executable prior to protecting with Sentinel LDK Envelope.

## Mandatory Protection Parameters

The following information must be provided in order to protect software using Sentinel LDK Envelope:

- > **Input file location:** You must specify the location of the program that you want to protect. By default, this is the directory from which you added the program to the project.
- > **Output file location:** You must specify the directory where the protected output will be saved. By default, the directory is:

*%UserProfile%\Documents\Thales\Sentinel LDK 10.0\VendorTools\VendorSuite\Protected*

- > **Vendor Code:** You must provide a valid Vendor Code in order to access a Sentinel protection key. On initial activation of Sentinel LDK Envelope, the default Vendor Code is specified as **DEMOMA**. Select your Vendor Code in the Sentinel Vendor Code screen.

This information is sufficient to protect a program.

## General Customizable Protection Parameters

The customizable parameters described in this section are identical for all supported applications, assemblies and dynamic libraries.

- > **Feature ID:** You can select a unique Feature to protect your program. For additional information about Features, see ["Using Features to Protect Programs" below](#).
- > **Protection key search mode:** You can determine where a protected application searches for the Sentinel protection key. For additional information, see ["Searching for a Sentinel Protection Key" on the next page](#).

**NOTE** When enabling or disabling some features you might reduce the level of protection provided by the software.

## Using Features to Protect Programs

A *Feature* is an identifiable functionality of a software application. Features may be used to identify entire executables, software modules, .NET or Java methods, or a specific functionality such as Print, Save or Draw. Each Feature is assigned a unique identifier called a Feature ID. The default Feature ID in Sentinel LDK Envelope is Feature ID 0.

For additional information on Features and licensing, see ["Identifying Functional Components \(Features\)" on page 100](#) and ["Managing Features" on page 109](#).

When you protect a Win32, Windows x64, Mac or Linux application with Sentinel LDK Envelope, you specify a single Feature ID for the entire executable. If you wish to apply unique Features to separate components or functionalities, you must use the Sentinel Licensing API. For additional information, see ["Sentinel Licensing API Protection" on page 43](#).

## Protecting .NET Assemblies

When you protect a .NET assembly with Sentinel LDK Envelope, you have the flexibility to specify Features at two levels:

- > A global Feature that relates to the entire .NET assembly, with the exception of individually-protected methods. For additional information, see ["Global Features in .NET Assemblies" on page 63](#).
- > Method-specific Features. For additional information, see ["Method-specific Features and Parameters in .NET Assemblies" on page 64](#).

At run-time, a protected .NET assembly searches for all Features in the Sentinel protection key.

## Searching for a Sentinel Protection Key

Sentinel LDK Envelope enables you to determine where a protected application searches for a required Sentinel protection key.

The following options are available:

- > **Local and remote:** The protected application first searches the local machine for a required Sentinel protection key (default), and then the network.
- > **Local only:** The protected application searches only the local computer for a required Sentinel protection key.
- > **Remote only:** The protected application searches only the network for a required Sentinel protection key.

## Sentinel LDK Envelope for Windows

---

This section describes how to use Sentinel LDK Envelope on Windows platforms.

*In this section:*

- > ["Prerequisites for Windows" below](#)
- > ["Running Sentinel LDK Envelope" on the next page](#)
- > ["Protecting Windows Programs" on the next page](#)
- > ["Calling Licensing API Functions From DllMain" on page 58](#)
- > ["Enhancing Protection With "AppOnChip"" on page 58](#)
- > ["Disabling Protection Keys When Tampering Is Detected" on page 60](#)
- > ["Accessing and Protecting Data Files" on page 61](#)
- > ["Running Sentinel LDK Envelope from a Windows Command Line" on page 61](#)

## Prerequisites for Windows

To use Sentinel LDK Envelope, all of the following components must be installed on your system:

- > Sentinel LDK Run-time Environment
- > Sentinel Vendor Suite
- > A valid Vendor Code stored in the **VendorCodes** folder. For additional information, see ["Extracting the Vendor Code from Sentinel Vendor Keys" on page 45](#).
- > `dfcrypt.exe` (if you are planning to encrypt data files by means of a command line)
- > The Win32, Windows x64, .NET or Java executables or DLLs that you want to protect
- > .NET Framework 2.0 or later (if you are protecting .NET assemblies)



## Running Sentinel LDK Envelope

From the Start menu, open **Sentinel LDK**. From the Sentinel LDK Launcher, select **Sentinel LDK Envelope**.

## Sentinel LDK Envelope Protection Parameters

After your program has been included in a Sentinel LDK Envelope project, protection can be performed effortlessly, based on the default Sentinel LDK Envelope settings. In addition, you can define and calibrate a range of protection parameters that affect the attributes and behavior of the protected application.

Sentinel LDK Envelope customizable parameters are displayed in the Protection Details screen and the Default Protection Settings screen. You can select a specific program in the Project pane and, from the Protection Details screen, view and edit the application's parameters using the following three tabs:

- > General tab
- > Advanced tab
- > Protection Settings tab

All parameters are detailed in the [Sentinel LDK Envelope for Windows](#).

This section provides an overview of the Sentinel LDK Envelope protection settings that are common to all program types. Mandatory parameters that are required in order to protect a program are described in ["Mandatory Protection Parameters" on page 54](#). Other common parameters are described in ["General Customizable Protection Parameters" on page 55](#).

Sentinel LDK Envelope also provides settings that are specific to the type of program protected.

- > For additional information about settings for Win32 or Windows x64 programs, see ["Protecting Windows Programs" below](#), and ["Accessing and Protecting Data Files" on page 61](#).
- > For additional information about settings for .NET assemblies, see ["Protecting .NET Assemblies" on page 62](#), and ["Code and Symbol Obfuscation in .NET Assemblies" on page 65](#).
- > For additional information about settings for Java executables, see ["Protecting Java Executables" on page 70](#).

## Protecting Windows Programs

When you protect a Windows program with Sentinel LDK Envelope, you can determine protection attributes and aspects of the behavior of the protected application.

## Protected Application Behavior

Sentinel LDK Envelope enables you to define the following additional properties for Win32 and Windows x64 programs:

- > The frequency at which random queries are sent to a Sentinel protection key. These queries include random encryption and decryption procedures.

- > The time interval between checks for the presence of a required Sentinel protection key.
- > Whether support for programs that require overlays to execute correctly should be enabled.
- > The length of time that the protected application waits for the Sentinel LDK Run-time Environment to load.

## Protection Attributes

You can define specific security attributes for protected Win32 and Windows x64 programs including parameters for:

- > Detection of both system and user-level debugging measures. You can activate measures to be undertaken by the Sentinel LDK system to block potential attacks intended to undermine the protection scheme.
- > Specifying the frequency of Sentinel protection key access for encryption. The parameter controls the compactness of the Sentinel protection key calls made by the protected application.

## Run-time User Support

You can customize run-time messages for end users who are using applications protected by Sentinel LDK Envelope. Sentinel LDK Envelope includes a set of message codes. Each code is mapped to a corresponding message that is displayed at run-time of the protected application.

In addition, you can choose to display a message for end users during startup of a protected application that explains there may be delays due to required data decryption.

## Calling Licensing API Functions From DllMain

In Windows and .Net, DLLs are loaded at run-time according to the order in which they were linked while building the application. During protection of the application or DLL with Sentinel LDK Envelope, the Licensing API is integrated into the protection application. The Licensing API sometimes calls functions that are external to KERNEL32 from DllMain, which is not recommended by Microsoft. This can cause the protected application to crash, because the protected DLL is loaded and executes before the required setup has been completed.

To prevent this occurrence, Thales recommends that you do one of the following:

- > Use the External License Manager (hasp\_rt.exe). When using the External License Manager, the Licensing API calls only KERNEL32 functions. This is the preferred solution, and this is the only solution that satisfies the Microsoft recommendation that you call only KERNEL32 functions from DllMain.
- > If using the External License Manager is not an option, add the protected DLL at the end of the import list of the application. This problem does not occur when the DLL is loaded with LoadLibrary() as the latest library.

For more information regarding the External License Manager, see ["Types of License Managers" on page 209](#).

## Enhancing Protection With "AppOnChip"

Sentinel LDK Envelope incorporates AppOnChip protection to significantly increase the security of an application that is protected with a Sentinel HL (Driverless configuration) key.

**NOTE** This section is relevant for native Windows applications. For information on protecting .NET assemblies with AppOnChip, see the [Sentinel LDK Envelope for Windows](#) help system.

Currently, the following limitations apply for the application to be protected using AppOnChip:

- > You cannot use AppOnChip to protect the Licensing API DLL.
- > AppOnChip protection cannot be applied to applications and DLLs that have already been protected with tools from other vendors or sources.

**NOTE** An application that is protected using AppOnChip is not compatible with Sentinel SL keys or with any HL keys other than Sentinel HL (Driverless configuration) keys. If no Sentinel HL (Driverless configuration) key is present at execution time, an application that is protected using AppOnChip will stop and display a message that asks the user to attach an appropriate HL key.

If the protected application will be licensed using a Sentinel HL Basic key or Sentinel HL Pro key, you must connect a Developer key that contains the AppOnChip module at the time that you protect the application. For more information, see ["AppOnChip Module" on page 316](#).

Once enabled, AppOnChip uses a code transformation engine to analyze the application code. AppOnChip searches the application code for code fragments that can be offloaded to the Sentinel HL key. Functions containing eligible code fragments are listed in a table on the **AppOnChip** tabbed page in the Sentinel LDK Envelope interface.

AppOnChip identifies eligible code fragments in a two-step process:

1. AppOnChip identifies the application's functions using a map file. If a map file is not available, AppOnChip uses only the functions exported by the exe/dll (exports) for its subsequent analysis. Currently, only map files generated by a Microsoft Visual Studio compiler or a Delphi compiler are supported.
2. AppOnChip analyzes the machine code of the functions identified in the first step and searches for code fragments that are suitable to be extracted and executed by the Sentinel HL key. Functions that contain suitable code fragments are added to AppOnChip's list of functions.

You can examine AppOnChip's list of functions and modify the selections to include only those functions that you want AppOnChip to protect.

AppOnChip also provides a *Performance Profiling* facility. This facility equips the eligible code fragments so that the application collects runtime statistics for these code fragments when they are executed. Minimal runtime overhead is added by this process. These statistics are then used by AppOnChip to help you balance application security and performance.

When Envelope generates the protected application, AppOnChip automatically removes the eligible code fragments from the selected functions and replaces them with a transition code. The extracted code fragments are encrypted and signed with a vendor-specific key, and saved as part of the protected application.

Note that with the supported compilers (described above), the protection process is fully automatic. It is not necessary for you to make any changes to your application code to accommodate this process.

At run-time, when the application calls one of the protected functions, the encrypted code fragment is uploaded to the Sentinel HL key. Within the key, the code is decrypted and loaded into a virtual machine. Once loaded, the code is executed by the virtual machine. The output of the code is passed back to the protected function so that the application can continue to execute.

As a result of this process, protected code fragments are never exposed in any manner that would enable a cracker to analyze or disassemble the code.

For more information regarding the AppOnChip functionality, see the [Sentinel LDK Envelope for Windows](#).

## Disabling Protection Keys When Tampering Is Detected

**NOTE** Tamper detection functionality is not supported for the DEMOMA Batch Code.

A native Windows application that is protected with Sentinel LDK Envelope can provide protection against tampering when licensed with one of the following keys:

- > Sentinel HL (Driverless configuration) key with firmware version 4.54 or later.
- > Sentinel CL key (Sentinel LDK Run-time Environment 8.51 or later is required on the user's machine)

The Envelope run-time module in the application can disable the key if the module determines that the user has attempted to tamper with the key or with the protected application. Once the key is disabled, the protected application will no longer execute. The application will fail or will display an error message, depending on the type of tampering detected.

In Sentinel LDK Envelope, tamper detection functionality can be enabled in a protected application by selecting the parameter **Disable key for attempted tampering** in the Protection Settings screen.

The "disabled" state of a key can be determined at the customer site by the `get_info` function in Sentinel Licensing API and can be viewed in Sentinel Admin Control Center.

You have the option to re-enable a disabled key. Obtain a C2V file from the customer and do one of the following:

- > Check the C2V file into Sentinel LDK-EMS. Click the **Enable key** function to generate a V2CP (vendor-to-customer package) file. This file will contain all pending V2C license updates from Sentinel LDK-EMS for the key, including a special V2C that re-enables the key. You or your customer can apply this file in the same manner that any V2C file is applied.

If you want to determine the reason that the key was disabled, send the reason code displayed when you checked in the C2V file to Thales Technical Support.

- > Use the C2V file in Sentinel License Generation API to generate a license update with `SNTL_LG_LICENSE_TYPE_CLEAR_DISABLED_STATE`. Your customer can apply the resulting V2C file to re-enable the key.

To determine the reason that the key was disabled, decode the C2V file and send the displayed reason code to Thales Technical Support.

The "disabled" state of the key does not affect your ability to send license updates to the key. Any updates that were applied to the key before or after the key was disabled will be in force if you re-enable the key.

## Accessing and Protecting Data Files

When you use Sentinel LDK Envelope to protect a Windows application, you can add the capability to access and write data to protected data files.

A given protected application can be equipped with either of two modes of data file protection:

- > **Version 1** - This is appropriate for general-purpose data files. In this mode, the contents of the data files are encrypted. The data files can only be accessed by applications that have been protected with the vendor's unique Vendor Code and that have been provided with a specific encryption key.
- > **Version 2** - This is appropriate for important data files that you want to license separately, such as training video files and courseware. In this mode, the data files are encrypted and protected by assigning a Feature ID. Only users that purchase the specific license required for the files are able to access the files.

For a complete description of the available data protection options, see ["Protecting Data Files" on page 77](#).

## Running Sentinel LDK Envelope from a Windows Command Line

Sentinel LDK Envelope can be initiated using a command-line prompt. This is useful when running automated processes that do not require a graphical interface.

**NOTE** The command-line version of Sentinel LDK Envelope is primarily used for automated processes. Before running the command-line utility, create and save protection projects using **envelope.exe**.

To access the command-line version of Sentinel LDK Envelope, go to:

*%ProgramFiles(x86)%\Thales\Sentinel LDK\VendorTools\VendorSuite\envelope.com*

To start the command-line version of Sentinel LDK Envelope, type `ENVELOPE` in the command line.

## Command-line Options

The table that follows describes parameters that are available for use with the command-line version of Sentinel LDK Envelope.

Command	Description
-h --help	Displays the list of command-line parameters. Press Enter to return to the command-line console.
-p <project> --protect <project>	The command-line utility uses the specified project as input data for the application-wrapping process—all the files included in the project are protected.

Command	Description
<project>	The command-line version starts the GUI version with the specified project running as the current project.

## Protecting .NET Assemblies

Sentinel LDK Envelope provides significant flexibility when protecting .NET assemblies. In addition to global protection settings that you specify using the **Protection Details** and **Protection Template Settings** functionalities, you can also specify Method-level protection, by defining individual methods in the .NET assembly.

**NOTE** For information on protecting .NET assemblies with AppOnChip, see the [Sentinel LDK Envelope for Windows](#).

You can also define protection settings in your source code using custom attributes.

For details about the prerequisites for protecting a .NET assembly, and other considerations to take into account, see [".NET Considerations" below](#).

When you protect a .NET assembly with Sentinel LDK Envelope, you specify a global Feature that protects the entire assembly. For additional information, see ["Global Features in .NET Assemblies" on the next page](#).

In addition to the global Feature, you can define Features for individual methods. You can also define other method-specific parameters. For additional information, see ["Method-specific Features and Parameters in .NET Assemblies" on page 64](#).

You can also apply different levels of obfuscation to your .NET assembly. For additional information, see ["Code and Symbol Obfuscation in .NET Assemblies" on page 65](#).

Envelope can protect mixed-mode .NET applications. Only managed code is protected.

### .NET Considerations

When protecting .NET assemblies, consider the following issues:

- > You must protect your assemblies in a development environment. Sentinel LDK Envelope requires libraries that are not part of the .NET framework, but are included in the development environment.
- > Sentinel LDK Envelope for .NET requires access to all assemblies and their dependencies.
- > Sentinel LDK Envelope breaks the strong name signature of signed assemblies. You can choose to re-sign the assembly in Sentinel LDK Envelope, as part of the protection process.
- > When you protect a .NET Framework 1.x assembly, the Sentinel LDK Envelope output is in Framework 2.0, requiring Framework 2.0 to be installed on the end-user machine.

- > For your protected .NET assembly to function at run-time, a Sentinel LDK DLL is required. For more information, see ["Sentinel LDK Run-time Environment" on page 189](#).

## Global Features in .NET Assemblies

When you protect a .NET assembly with Sentinel LDK Envelope, you specify a global Feature that protects any methods that have not had individual protection parameters applied. The global Feature is also used when background checks are implemented.

## Method-level Protection

When you select a .NET assembly for protection, Sentinel LDK Envelope automatically determines the **Protection type** that will provide the best protection for your program, depending on whether you are protecting an executable or a DLL. The **Protection type** determines the methods that are available for individual protection.

**NOTE** It is recommended that you do not change the automatic **Protection type** settings.

This section describes how you select individual methods and the behavior of different method types, in addition to the parameters you can select for the methods.

## Selecting .NET Methods for Protection

The .NET assembly is displayed in the Protection Details screen, in the **Methods selected for protection** list. The list displays class constructors and methods, in a tree layout that mimics the structure of the .NET assembly.

Items in the list are identified by icons that indicate the method type, and by the class or method name. Method signatures are displayed as a tool tips.

When the check box to the left of a method is selected, that method is selected for Sentinel LDK Envelope protection.

### NOTE

- > Selecting or clearing the check box of a higher-level item does not affect nested items. For example, if you clear the check box of a class constructor, methods nested under it remain selected.
- > When a method name is grayed-out, it cannot be selected for protection.
- > If the **Protection type** is **Only Windows shell**, you cannot protect individual methods in that .NET assembly.
- > An assembly cannot be protected when the check boxes for all items in the list have been cleared.

## Method-specific Features and Parameters in .NET Assemblies

You can use Sentinel LDK Envelope to define separate Feature IDs for individual methods in your .NET assembly. This enables you to:

- > Make use of the separate encryption key inherent in each Feature to provide enhanced security for individual methods
- > Determine how often the protected application logs into an individual method

At run-time, the protected application searches for all relevant Feature IDs in the Sentinel protection key.

You can determine how often the protected application logs into each Feature ID in the Sentinel protection key and performs decryption using that Feature ID by specifying the **Frequency** for specific methods.

### NOTE

- > You can only specify the **Feature ID** and **Frequency** for methods that have been selected for protection.
- > If the **Protection type** is **Only Win32 shell** or **Only Windows x64 shell**, you cannot specify a **Feature ID** or **Frequency** for individual methods.
- > You can select multiple methods and specify the same **Feature ID** and **Frequency** for all selected items.

The available **Frequency** options are described in the following table:

Frequency Type	Description
<b>Once per program (Default)</b>	A check is performed the first time a method using the Feature ID indicated for that method is called, regardless of the number of methods that share the same Feature ID across the program.
<b>Once per class instance</b>	<p>A check is performed when the method is run, once for each Feature ID within the same class.</p> <p>If the same Feature ID is also assigned to the class constructor, the check is performed the first time the .ctor method is run.</p> <p>If the same Feature ID is used in other classes, the check is performed separately for each class.</p> <div> <b>NOTE</b> The Once per class instance frequency is available only for Instance methods.         </div>
<b>Every time</b>	A check is performed every time the method is called.



## Recommendations:

- > Use the **Once per Application** default setting. The **Once per Instance** and **Every time** settings may slow the performance of your program.
- > If an execution-based license is being defined, use the **Every time** setting only for the method that determines licensing, as the counter is decremented every time the method is called.

If you choose to assign separate Feature IDs for individual methods, you must ensure that your application code can only call the Feature IDs for those methods for which a valid license has been installed in a Sentinel protection key.

If methods that do not have a valid license in a Sentinel protection key are called, it will cause Sentinel LDK Envelope to generate an error loop that can only be stopped by installing a valid license.

An API is provided as part of the Sentinel LDK installation that enables you to ensure that the error loop does not occur. The *.NET Envelope Runtime API* is located in:

```
%ProgramFiles(x86)%\Thales\Sentinel LDK\Samples\
Envelope\EnvelopeRuntime.NET
```

For information on using this API, see *.NET\_Envelope\_Runtime\_API.html* in the above location.

## Code and Symbol Obfuscation in .NET Assemblies

Obfuscation is the process of turning meaningful strings into random strings of letters or numbers. Using Sentinel LDK Envelope, you can apply obfuscation as an anti-reverse engineering security measure.

By default, all symbol names in the protected .NET assembly are obfuscated as part of the protection process. In addition, you can choose to obfuscate the entire code of a selected method. Since code obfuscation may slow the performance of your program, it is not selected by default.

You can apply Code obfuscation to a method regardless of whether it is selected for protection in the **Methods selected for protection** list.

## Exception Handling in Protected .NET DLL Assemblies

Given the following circumstances:

- > A .NET DLL assembly is protected with Sentinel LDK Envelope.
- > An application (protected or unprotected) calls the .NET DLL assembly.
- > An appropriate license for the .NET DLL assembly cannot be located.

The .NET DLL assembly will raise a system exception. This is the expected behavior. The application that called the .NET DLL assembly should contain appropriate code to catch the exception and handle it. For example, the application can display a message stating that the user must connect an HL key that contains the required license.

Note that for an EXE assembly, the code added by Sentinel LDK Envelope can notify the user that the required protection key was not found and can quit the program because this issue is detected at program startup. But a .NET DLL assembly can be called at any point during a program's execution, so it is up to the developer to decide how to respond if the required license is not found.

## Defining Sentinel LDK Envelope Protection Settings in Source Code

Instead of specifying your protection settings using the Sentinel LDK Envelope GUI, you can use the .NET framework custom attributes for the `Aladdin.HASP.Envelope` assembly to add definitions directly to your source code.

The custom attributes can be applied to assemblies, classes, and methods. Protection settings in your source code are processed according to hierarchy, in descending order of method, class, and assembly.

For more information, see *.NET\_Envelope\_Configuration\_API.html*. This document can be found in the following location:

```
%ProgramFiles(x86)%\Thales\Sentinel LDK\Samples\
Envelope\EnvelopeRuntime.NET
```

## Protecting Python Applications

The following methods exist to protect Python applications using Sentinel LDK Envelope on a Windows or Linux machine:

### 1. Script Envelope

Script Envelope is an advanced command-line tool for applying Sentinel LDK Envelope protection to Python applications.

**NOTE** Due to its built-in automation, Script Envelope is the simplest and preferred method for applying Sentinel Envelope protection to your applications.

### 2. Cython (<https://cython.org/>) followed by Sentinel LDK Envelope

This method protects Python applications by combining Cython with Sentinel LDK Envelope. This works by first translating your sensitive Python modules into native modules (PYD/SO files) which are then protected using Envelope.

This method provides a higher level of security because the additional compilation step lowers the code's abstraction level and enables Sentinel LDK Envelope to protect the application as code and not just as data. This results in more sophisticated protection measures. However, this method is slightly more complicated to set up, as it requires Cython and a working C compiler.

**NOTE** This method cannot protect an application's start script, only its Python modules. Therefore, Thales recommends that you place your application's actual entry point inside a Python module and only use the start script to call the module.

You can perform either protection process using Sentinel LDK Envelope on a Windows or Linux platform. For more information, see the [Sentinel LDK Envelope for Windows](#) or the [Sentinel LDK Envelope for Linux](#).

## Sentinel LDK Envelope for Linux Applications

---

Sentinel LDK Envelope protection can be implemented for Linux executables and shared objects using a command-line utility.

The Sentinel LDK Envelope command-line utility runs on a Linux Intel platform. However, it can be used to protect both Linux Intel and Linux ARM applications.

For a complete description of the Sentinel LDK Envelope command-line utility, see the [Sentinel LDK Envelope for Linux](#). This guide can be found under `\Linux\Docs\Manuals & Tutorials\` where Sentinel LDK files for Linux are installed.

When you use Sentinel LDK Envelope to protect a Linux application, Envelope adds the capability to access data from protected data files. The customer must possess a license for the appropriate Feature in order to access the protected data files.

**NOTE** A protected Linux application cannot modify data in a protected data file.

You can pre-encrypt and assign licensing parameters to data files that you will deliver together with the protected application. The following utilities can be used for this purpose:

- > Sentinel LDK Data Protection utility (under Windows)
- > **dfcrypt** utility (under Linux or Windows)

For Linux applications, only the **Version 2** mode of data file protection is supported. For a complete description of the available data protection options, see "[Protecting Data Files](#)" on page 77.

## Sentinel LDK Envelope for Mac Binaries

---

Sentinel LDK Envelope for Mac enables you to protect Mach-O executables and dynamic libraries (referred to as binaries). Both GUI and command-line versions of the application are available.

Before using Sentinel LDK Envelope for Mac, it is recommended that you familiarize yourself with the general Sentinel LDK Envelope information about Sentinel LDK Envelope protection that is provided at the beginning of this section.

### Sentinel LDK Envelope Prerequisites for Mac

To use the Sentinel LDK Envelope utility, all of the following components must be installed on your system:

- > Sentinel LDK Run-time Environment
- > Sentinel Vendor Suite, containing the Sentinel LDK Envelope and the Master Wizard

- > A valid Vendor Code stored in the **VendorCodes** folder. For additional information, see ["Extracting the Vendor Code from Sentinel Vendor Keys" on page 45](#).
- > The Mac binaries that you want to protect

## Running Sentinel LDK Envelope for Mac

Navigate to the location in which Sentinel LDK is stored. Select **MacOS > VendorTools > VendorSuite > Envelope**. Sentinel LDK Envelope is launched.

To access the command-line version of Sentinel LDK Envelope, go to:

`...MacOS\VendorTools\VendorSuite\envelope_darwin`

Type `envelope_darwin -h` in the command line to start the command-line version of Sentinel LDK Envelope.

## Sentinel LDK Envelope for Mac Protection Parameters

After your Mac executable or dynamic library has been included in a Sentinel LDK Envelope project, protection can be performed effortlessly, based on the default Sentinel LDK Envelope settings. In addition, you can define and calibrate a range of protection parameters that affect the attributes and behavior of the protected binary.

Sentinel LDK Envelope customizable parameters are displayed in the Protection Details screen and the Default Protection Settings screen. You can select a specific binary in the Project pane and, from the Protection Details screen, view and edit the binary's parameters using the following three tabs:

- > General tab
- > Advanced tab
- > Protection Settings tab

All parameters and procedures are detailed in the [Sentinel LDK Envelope for Mac](#).

## Accessing and Protecting Data Files

When you use Sentinel LDK Envelope to protect a Mac application, you can add the capability to access and write data to protected data files.

A given protected application can be equipped to create, access, and update protected data files. The data files can only be accessed by applications that have been protected with the vendor's unique Vendor Code and that have been provided with the encryption key that was used to protect the files.

You can use Sentinel LDK Data Protection utility to pre-encrypt data files that you want to deliver together with the protected application.

For a complete description of the available data protection options, see ["Protecting Data Files" on page 77](#).

## Sentinel LDK Envelope for Java Executables

Sentinel LDK Envelope for Java enables you to protect JAR and WAR executables. Before using Sentinel LDK Envelope for Java, it is recommended that you familiarize yourself with the general Sentinel LDK Envelope information about Sentinel LDK Envelope protection that is provided at the beginning of this section.

Protection of your software is performed on a Windows machine, after which you distribute the protected software together with the appropriate Java run-time libraries for the end-user operating system—Windows, Mac, or Linux.

**NOTE** Java applications that have been obfuscated, or protected using third-party tools, are not supported by Sentinel LDK Envelope.

### Java Considerations

When protecting Java executables, consider the following issues:

- > The methods selected for protection by Sentinel LDK Envelope by default are not the optimal choices for your application or library. You must review and modify the list of selected methods to provide the best mix of security and performance. For more information, see the description of optimizing protection settings in the [Sentinel LDK Envelope for Windows](#).
- > Sentinel LDK Envelope does not support protection of Java paint methods, but it allows you to select them in the user interface. As a result, the protected application may cause a deadlock when it executes a protected paint method at runtime with no Sentinel protection key connected. To prevent this issue from occurring, you can deselect all paint methods. Note that paint methods do not usually contain application logic; therefore, deselecting them typically has no impact on security. As an alternative, you can select console output for messages by enabling *stderr* output instead of *windows* in the Advanced settings panel.
- > When you test Sentinel LDK Envelope for the first time with your application, it is recommended that you clear the default selection and start with the protection of a single method that you want to protect. After you protect the method, test your application. If the application works as expected, continue to protect additional methods and test after each addition until you have reached the desired protection selection for the application. Do not try to apply this selection to different applications.
- > Sentinel LDK Envelope does not support protection of methods that use the Hibernate service.
- > Sentinel LDK Envelope does not support protection of methods that, in turn, use Synthetic methods that are created as bootstrap methods or as arguments of bootstrap attributes.

### Sentinel LDK Envelope Prerequisites for Java

To use the Sentinel LDK Envelope for Java engine, all of the following components must be installed on your system:

- > The Java JRE or JDK
- > Sentinel LDK Run-time Environment

- > Sentinel Vendor Suite, containing the Sentinel LDK Envelope and the Master Wizard
- > A valid Vendor Code stored in the **VendorCodes** folder. For additional information, see ["Extracting the Vendor Code from Sentinel Vendor Keys" on page 45](#).
- > The JAR or WAR executables that you want to protect

Before your JAR/WAR archive is protected, include the following customized Sentinel Licensing API dynamic libraries with the archive:

Operating System	Customized Sentinel Licensing API Dynamic Libraries
Windows (32/64-bit)	hasp_windows_****_<vendorId>.dll
ARM	hasp_linux_arm64_<vendorId>.so and hasp_linux_armhf_<vendorId>.so
Linux (32/64-bit)	libhasp_linux_***_<vendorId>.so

During protection of Java applications, Sentinel LDK Envelope copies these libraries automatically to the output directory from the default location (generated after the Master key or Developer key is introduced with Master Wizard).

**NOTE** Envelope does not copy vendor-specific Mac files automatically to the output directory. You need to copy hasp\_darwin\_<vendorId>.dylib with the protected archive manually.

For your protected Java executables to function at run-time, one or more Sentinel LDK DLLs are required. For more information, see ["Sentinel LDK Run-time Environment" on page 189](#).

## Running Sentinel LDK Envelope for Java Engines

From the Start menu, open **Sentinel LDK**. From the Sentinel LDK Launcher, select **Sentinel LDK Envelope**.

## Sentinel LDK Envelope for Java Protection Parameters

After your Java executable has been included in a Sentinel LDK Envelope project, protection can be performed, starting from the default Sentinel LDK Envelope settings. In addition, you can define and calibrate a range of protection parameters that affect the attributes and behavior of the protected file.

Sentinel LDK Envelope customizable parameters are displayed in the Protection Details screen and the Default Protection Settings screen. You can select a specific Java executable in the Project pane and, from the Protection Details screen, view and edit its parameters using the available tabbed pages.

## Protecting Java Executables

When you protect a Java executable with Sentinel LDK Envelope, you can determine protection attributes and aspects of the behavior of the protected application.

## Protected Application Behavior

Sentinel LDK Envelope enables you to define the following additional properties for Java executables:

- > The compression level of protected classes.
- > The time interval between checks for the presence of a required Sentinel protection key.

All parameters and procedures are detailed in the [Sentinel LDK Envelope for Windows](#) Sentinel LDK Envelope.

## Defining Sentinel LDK Envelope Protection Settings in Source Code

Protection settings for an application are typically specified by using the Sentinel LDK Envelope user interface.

For certain settings, the *Java Envelope Configuration API* provides an alternate method for applying protection. Using this API, the developer can specify protection settings for methods directly in the application's source code.

For more information, see *Java\_Envelope\_Configuration\_API.html*. This document can be found in the following location:

```
%ProgramFiles(x86)%\Thales\Sentinel LDK\Samples\  
Envelope\EnvelopeRuntimeJAVA
```

# CHAPTER 6: Protection Strategies

Sentinel LDK provides the best hardware and software tools available in the market today. The contribution that Sentinel LDK can make to the protection of your software and intellectual property has already been well documented in the previous sections. However, it is the strength and sophistication of the strategies that you employ in partnership with Sentinel LDK that will truly maximize your software protection.

*In this section:*

- > ["Protection Strategies Overview" below](#)
- > ["General Protection Guidelines" on the next page](#)
- > ["Types of Attack and Their Sentinel LDK Defense" on the next page](#)

## Protection Strategies Overview

Parallel with advances in software and software security development, software crackers are developing more sophisticated means of deconstructing software protection measures—in order to duplicate and distribute illegal copies of unlicensed software—and to reverse engineer code in order to steal intellectual property.

To maintain the rights to your revenue stream, it is essential that you remain vigilant about the strategies of your “enemies”, and that you continually and wisely implement the latest and strongest techniques for protecting your software.

The degree of investment that you make in limiting the ability of software crackers to illegally access your software will depend on a number of considerations, including:

- > The value of your software
- > The history of previous cracking attempts related to your software
- > The geographical region in which your software will be distributed
- > The target market for your software (for example, whether it is intended to be sold to individual consumers, small office/home office users, or enterprise users)

There is no software protection that is absolutely uncrackable. However, if you constantly implement up-to-date strategies using the strongest software protection methods, you significantly decrease your vulnerability to such attacks.

This section describes general protection strategies for software vendors. It then outlines some of the methods that software crackers employ in order to identify and negate software protection and security, and recommends Sentinel LDK measures that you can use to enhance your software security.



In addition to the information described in this manual, our team of Thales Consultants provides personalized assistance in strengthening software security and protection. They can provide help on a wide range of issues, including additional protection strategies and implementation techniques.

For information on consultation services offered by Thales, contact your local Thales representative.

## General Protection Guidelines

---

The following guidelines should be followed, regardless of the software protection strategies being implemented.

Thales thoroughly and constantly investigates potential and actual threats to software security, and Sentinel LDK is continuously being updated to counter such threats—before they can compromise the security of your software.

### Use the Most Up-To-Date Protection Software

Protection software updates generally include enhancements to counter the most recent threats. Always check for and use the most recent version of Sentinel LDK protection software that is available. The latest software can be downloaded from the [Sentinel Web site](#).

### Constantly Re-Evaluate Protection Strategies

Frequently consider what protection strategies you can upgrade or enhance to provide stronger security for your software.

### Use Evolving Strategies to Prevent Predictability

Vary the strategies that you implement between your software releases. If a software cracker is able to detect a pattern to your protection strategies, the strategies can more easily be negated or evaded.

### Vary Behavior When a Cracking Attempt Is Detected

When a cracking attempt is detected (for example, through using a checksum—described later in the section), delay the reactive behavior of your software, thus breaking the logical connection between “cause” and “effect.” Delayed reaction confuses a software cracker by obscuring the link between the cracking attempt and the negative reaction of the software to that attempt.

Behavior such as impairing program functionality when a cracking attempt is detected can be very effective. Additional behaviors could include causing the program to crash, overwriting data files, or deliberately causing the program to become inaccurate, causing the program to become undependable.

## Types of Attack and Their Sentinel LDK Defense

---

It is important to “know your enemy.” When you are well informed about the types of attacks that a software cracker may make, you will be best able to devise and implement strategies that limit or prevent their success.

This section describes the elements of some of the more common attacks that software crackers use, and refers you to specific Sentinel LDK strategies that you can implement to counter such attacks.

## Patching Executables and DLLs

A software cracker disassembles and/or debugs EXE or DLL files to find protected code. The actual file is then patched in order to modify run-time flow, or to remove calls in the code.

Commonly, the software cracker sends a small, standalone patch executable that the end user runs in order to patch your software.

### Sentinel LDK Solution

The greater the number of protected files, the longer it takes a software cracker to remove protection. You can protect multiple executable and DLL files using Sentinel LDK Envelope. You can also use the Data Protection facility to encrypt and protect data files that are accessed by protected applications.

## Modifying Key Memory

Licensing data is normally stored in the memory of a software protection key. A software cracker attempts to access the Protection Key memory in order to modify the licensing terms. For example, a depleted execution-based license might be changed to a perpetual license, or a feature that has not been paid for might be enabled.

### Sentinel LDK Solution

In the context of Sentinel LDK, Read-only memory (ROM) is a segment of the memory that can contain data that the protected application can access, but cannot overwrite. Sentinel protection keys contain two ROM segments, one of which contains Sentinel LDK Feature-based licenses. The second segment provides an area in which vendor-customized data can be stored. These segments can only be updated using remote updates.

Sentinel LDK automatic Feature-based licenses utilize read-only memory of Sentinel protection keys. The different types of available licenses are sufficient for almost any licensing model.

You can customize your own licenses and still use a ROM segment in a Sentinel protection key's memory. Note however that licenses that have been customized must remain static (for example, such licenses cannot include a decremented number of executions).

For additional information about licensing models, see ["Part 5 - Licensing Business Models" on page 239](#).

## Emulating Protection Keys

To emulate the software of a protection key manufacturer, a software cracker creates an application that replays previously recorded calls, as if an actual protection key is returning the calls.

Limited functionality emulators only record and replay calls. Full-functionality emulators also emulate the key, including its encryption. A software cracker requires access to the encryption key to create a full-functionality emulator.

There are several places in which emulators can reside. Primarily, they are an attempt to replace the driver.

### Sentinel LDK Solution

Sentinel LDK provides a secure channel between an application and the Sentinel HL key. Data that passes between the protected application and the key is encrypted. Taking advantage of the secure channel functionality between your application and a Sentinel HL key provides you with the strongest possible protection.

A different encryption key is used in every session. This means that someone recording data passing through the secure channel cannot replay the data, since the encryption key used to encrypt the data will differ from that used to decrypt the data.

## Using Remote Desktops and Remote Desktop Solutions

When using the remote desktops of some operating systems, it might be possible for an end user with a standalone protection key to enable software on multiple remote desktops simultaneously.

### Sentinel LDK Solution

The Sentinel LDK protection includes mechanisms to determine if a protected application is running on a remote desktop. If such a situation is detected, and a Feature in the license is not specifically enabled for remote desktops, the program will not function.

## Cloning Hardware Keys

The software cracker reverse-engineers a hardware protection key, then creates duplicates. Such an attack is extremely costly to the cracker, both in terms of the reverse engineering tools and the expertise required. It is also costly in terms of ongoing production of hardware keys.

### Sentinel LDK Solution

Sentinel HL keys are each unique and have their own ID. Keys that are in the same Batch Code and behave identically are each uniquely encrypted, the key's customized controller and memory forming a unique locked pair. This means that if the memory of one Sentinel HL key is copied to another Sentinel HL key, the second key will not function.

## Clock Tampering

Clock tampering relates to either the system clock of the machine on which the protected software is running, or to a real-time clock contained in keys. The software cracker resets the time to enable extended, unlicensed use of the software.

### Sentinel LDK Solution

When implementing time-based licenses for your software, use one of the following keys:

- > Sentinel HL Time or Sentinel HL NetTime keys. These keys provide a real-time clock.
- > Sentinel HL (Driverless configuration) key. This key provides a virtual clock (V-Clock). For more information, see ["How Sentinel LDK Protects Time-based Licenses With V-Clock" on page 346](#).

Both the clock itself, and the license which is stored in read-only memory, cannot be modified.

## Additional Sentinel LDK-specific Strategies

This section describes additional general protection strategies that are available to users of Sentinel LDK.

### Use Both the Sentinel Licensing API and Sentinel LDK Envelope

Maximize security by using the Sentinel Licensing API to implement calls to a Sentinel protection key, and protect the application with Sentinel LDK Envelope. Using one protection method does not preclude the use of the other.

### Insert Multiple Calls in your Code

Inserting many calls, throughout the code, to the Sentinel protection key in order to check the presence of the key, and binding data from the key with the software functionality, frustrates those attempting to crack your software. Multiple calls increase the difficulty in tracing a protection scheme.

You can also add obstacles to a potential software cracker's progress by encrypting data that has no bearing on the application. Similarly, you can divert attention by generating "noise" through random number generators, time values, intermediate results of calculations, and other mechanisms that do not lead to meaningful results or actions.

### Encrypt/Decrypt Data with a Sentinel protection key

Encryption and decryption processes are performed inside a Sentinel protection key, well beyond the reach of any debugging utility.

Encrypting data with the Sentinel LDK AES-based encryption engine considerably enhances software security. By encrypting data used by your application, the decryption process depends on both the presence of a Sentinel protection key and its internal intelligence.

By implementing a Sentinel Licensing API scheme in which data is decrypted by a Sentinel protection key, the association between the protected application and the Sentinel protection key cannot easily be removed. Cracking the software also necessitates the software cracker decrypting the data.

### Use a Checksum to Verify Integrity of Executable Files

Compare the value in the executable file with a checksum stored in Sentinel protection key memory. If the two values are not equal, you can assume that someone has attempted to modify the files. Repeat this check in various places in the code, varying it in each place to make it more difficult for a software cracker to detect.

**NOTE** This strategy is not necessary if you protect your application with Sentinel LDK Envelope. Envelope implements its own integrity checks and uses code encryption to prevent modification of the protected application.

# CHAPTER 7: Protecting Data Files

This section describes how you can use Sentinel LDK Data Protection utility to protect data files.

*In this section:*

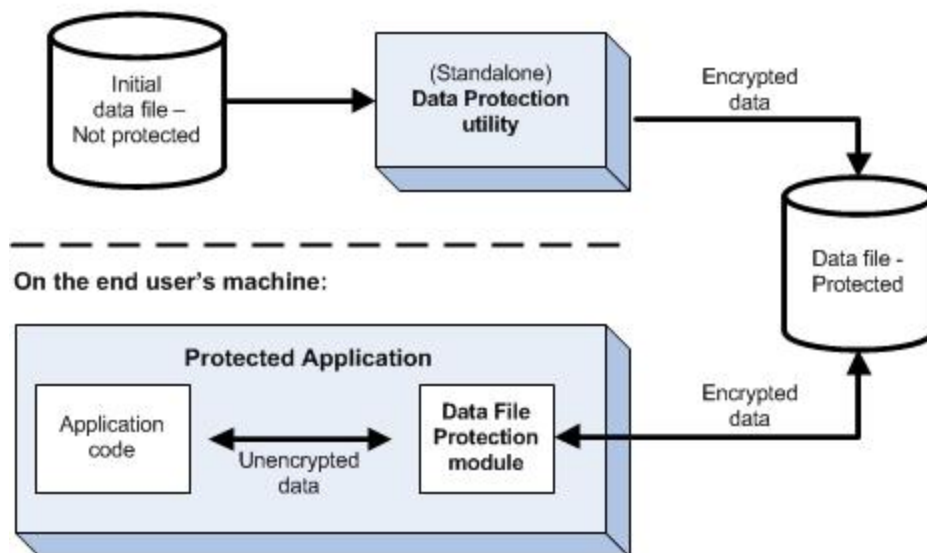
- > ["Overview" below](#)
- > ["Data File Protection Prerequisites " on page 80](#)
- > ["Launching Sentinel LDK Data Protection Utility" on page 81](#)
- > ["Licensing Data Files—Getting Started" on page 82](#)
- > ["Working With the dfcrypt Command Line Utility" on page 83](#)

## Overview

The material contained in a data file can represent a significant investment in time, effort, and money. For example, a data file may contain valuable text-based, audio, or video courseware for a training program.

Sentinel LDK provides you with a *Data Protection facility* to encrypt and (optionally) add licensing protection to the contents of data files, similar to the licensing protection that is available for software applications.

**At the vendor's development site:**



The Data Protection facility consists of the following components:

> **Sentinel LDK Data Protection utility**

This utility is used to protect data files that will be delivered together with a protected application or as separate files. The utility can be invoked from within Envelope (under Windows and Mac) or as a standalone application. The utility does the following:

- The utility encrypts the data file. Once encrypted, the file can only be accessed by one of the modules described below.
- The utility optionally assigns a Feature ID to the data file. If this is done, the data file can only be accessed if an appropriate protection key is available.

The utility is available as a GUI-based application (for Windows and Mac) or as the command-line utility **dfcrypt** (for Windows and Linux).

> **Data File Protection module**

This module is (optionally) inserted into the protected application by Sentinel LDK Envelope. This enables the protected application to access the data in a protected data file. If the data file has been protected using the **Version 2** protection mode (described below), the data file can only be accessed if an appropriate protection key is available.

The Data File Protection module can only be inserted into a protected executable file or DLL file. The module cannot be inserted into any other library file.

A protected application with the Data File Protection module can work with both protected data files and regular data files.

Both the Sentinel LDK Data Protection utility and the Data File Protection module provide two distinct modes of operation:

> **Version 1 (previously DataHASP)**

In this mode, data files that are created by or accessed by a protected application can be encrypted and decrypted by the Data File Protection module in the protected application. However, there are no specific license requirements to access the data files.

If you want to deliver data files together with the protected application, you can use the Sentinel LDK Data Protection utility to encrypt these files.

The protected data files that can be accessed by a protected application are managed by setting up the following controls in Sentinel LDK Envelope:

- **Data filters** - File masks that set rules to determine the names and file types of protected files that the protected application can access.
- **Data encryption key** - An eight-character key used to add an extra layer of encryption for protected data files. The same key must be provided in Sentinel LDK Envelope for each protected application that will

access a given protected data file or collection of protected data files. This key is also used by the Data Protection utility to encrypt the data files.

**Version 1** is supported for data files to be accessed under Windows, .NET (Windows shell), or Mac.

## > Version 2

In this mode, you can both encrypt and license data files with the Sentinel LDK Data Protection utility. Each data file or group of data files is assigned a specific Feature ID. To access the data file, the end user requires a protection key with a license for the relevant Feature ID. By distributing the relevant Feature IDs among various Products, you can easily manage the licensing of a large collection of data files.

This mode is especially suited for educational data and courseware. Data files protected in this mode are protected against video capture software that runs on the machine where the user is viewing the protected video file.

The data file can be accessed and modified by a protected application with the Data File Protection module (**Version 2**). The application and the data file must be protected with the same Batch Code.

With **Version 2** mode, the protected application cannot create a new protected data file. However, you can manually create an empty data file, protect it with the Sentinel LDK Data Protection utility, and deliver the file together with the protected application. The protected application can add content to the protected data file.

**NOTE** Under Linux, files protected with **Version 2** data protection mode are read-only.

For protected data files that are accessed using a protected application, the locking type for the data files is identical to the locking type for the application.

**Version 2** is supported for data files to be accessed under Windows, .NET (Windows shell), or Linux (Intel or ARM).

## When to Protect Data Files

Protect your data files if:

- > You want to maximize your software's security. When your software is being protected, consider adding another layer of security by protecting those data files that are accessed by your software.
- > You want to protect your intellectual property. Your data files may represent a significant investment, so it is worthwhile preventing your intellectual property from being exposed without protection.
- > You want to license your data files. You can assign a different Feature ID to each data file or to a group of data files. By distributing the relevant Feature IDs among various Products, you can easily manage the licensing of a large collection of data files.

## Users of Sentinel LDK Data Protection Utility

Anyone involved in the production or maintenance of data files for your protected software should use Sentinel LDK Data Protection utility. This could include people in roles such as graphic artists, information developers, or accountants.

## Data Encryption for Mac

Sentinel LDK Envelope for Mac provides the capability for a protected application under Mac OS X to encrypt and decrypt data that is written to and read from an external file.

Data files that will be delivered together with the protected application must be pre-encrypted using the Sentinel LDK Data Protection utility for Mac or for Windows.

When using Sentinel LDK Data Protection utility to encrypt files for Mac:

- > The Data Protection utility ignores TYPE/CREATOR for files.
- > The Data Protection utility does not work with document types stored in File Bundles (for example: Keynote presentations) since these are directory structures and not typical files.

## Data File Protection Prerequisites

The requirements for using data file protection are described in the tables that follow.

**NOTE** For supported platforms for vendors and for customers, see the [Sentinel LDK Release Notes](#).

## Requirements for Vendors

	Version 1 Data File Protection	Version 2 Data File Protection
<b>Main purpose of Data File protection</b>	To protect data files that are accessed by one or more protected applications. The data files are not licensed.	To protect and license data files that are accessed by one or more protected applications.
<b>Sentinel LDK Envelope</b>	Before protecting data files, you must create a Sentinel LDK Envelope project containing one or more programs for which data protection has been enabled and data filters have been defined. The data filters must include the data files that you want to protect.	Create a Sentinel LDK Envelope project containing one or more programs for which data protection has been enabled.



	Version 1 Data File Protection	Version 2 Data File Protection
<b>Sentinel LDK Data Protection utility (or dfcrypt utility)</b>	Optional. Only required if you want to provide encrypted data together with the protected application.	Required to protect and assign Feature IDs for licensing the data files.
<b>Vendor key requirements</b>	To work with Envelope or with the Data Protection utility, you must connect a Developer key to your machine.	
<b>Data encryption key</b>	If two or more protected applications will access a given protected data file, the same data encryption key must be defined in Envelope for each application.	Not relevant

## Requirements for Customers

	Version 1 Data File Protection	Version 2 Data File Protection
<b>Types of platforms that support protected data files</b>	Windows, .NET (Windows shell), Mac	Windows, .NET (Windows shell), Linux (Intel or ARM)
<b>License requirements</b>	Not applicable	The license must contain the Feature ID that was used to protect the data file.

## Launching Sentinel LDK Data Protection Utility

You can launch Sentinel LDK Data Protection utility as follows:

- > Directly from Sentinel LDK Envelope (under Windows or Mac).
- > From the Start menu, open **Sentinel LDK**. From the Sentinel LDK Launcher, select **Additional Tools > Sentinel LDK Data Protection Utility**.
- > (Windows) Click the `datahasp.exe` file, located in the following directory on your system:  
`%ProgramFiles(x86)%\Thales\Sentinel LDK\VendorTools\VendorSuite\`
- > (Mac) Click the `DataHASP` file, located in the following directory on your system:  
`MacOS\VendorTools\VendorSuite\DataHASP.app\Contents\MacOS\DataHASP`

(To use the **dfcrypt** command line utility, see ["Working With the dfcrypt Command Line Utility" on page 83.](#))

## Licensing Data Files—Getting Started

This section demonstrates how to get started with ["Licensing Data Files to be Accessed Using a Proprietary Application" below](#).

Instructions are provided for working with the HL Demo key.

**NOTE** If you prefer to work with your own Batch Code (or if you do not have an HL Demo key), prepare an HL or SL key that contains a license for Features 0 and 42. Use this key instead of the Demo key where required. Where the Vendor Codes file is required, use your unique Vendor Codes file instead of the DEMOMA Vendor Codes file. Connect your Developer key (with the required module) to your machine.

It is assumed that you already have a basic familiarity with Sentinel LDK. If not, perform the lessons provided in the *Sentinel LDK Software Protection and Licensing Tutorial*.

### Licensing Data Files to be Accessed Using a Proprietary Application

This procedure demonstrates how to prepare a data file to be licensed and accessed with your own application.

For this procedure, you will use a text viewer application (**TextViewer.exe**), provided by Thales, to represent your proprietary application, and a simple text file to represent the data file that you want to protect. The text viewer will be licensed with Feature ID 0 and the data file will be licensed with Feature ID 42. (Licenses for these Features are already present on the Demo key.)

#### To protect the text viewer application and the data file:

1. Using the Windows **notepad** application, prepare and save two copies of a text file that contains the name of your organization (or any other text string). Name the files **test\_42.txt** and **test\_99.txt**.
2. Do the following in Sentinel LDK Envelope:
  - a. Add the **TextViewer.exe** application to a new Sentinel LDK Envelope project. This application can be found in:
 

```
%ProgramFiles(x86)%\Thales\Sentinel LDK\VendorTools\VendorSuite\samples\
```
  - b. In the Project pane, select the **TextViewer** program.
  - c. On the **General** tabbed page, select the **Enable data file protection** check box.
  - d. In the version list box, select **Version 2**.
  - e. Click **Protect** to protect the application. The application is protected and licensed with Feature ID 0. Be sure to note the location where the protected application is saved.
  - f. Close the Protection Status box.
  - g. Save the Envelope project.

- h. On the **General** tabbed page, click **Encrypt Data**. Sentinel LDK Data File Protection utility is launched. The **Batch Code** field displays **DEMOMA**.
3. Do the following in Sentinel LDK Data File Protection utility:
  - a. From the menu bar, select **Actions > Add Files**.
  - b. In the Files to Encrypt dialog box, click **Add**.
  - c. Browse to the text files that you prepared in step 1 above. Select the **test\_42.txt** file. The file now appears in the Files to Encrypt dialog box.
  - d. Set the **Feature ID** field on the right side of the box to **42**.
  - e. Set the output directory to the location where the protected text file should be written.
  - f. Click **OK**. The dialog box closes, and the text file is listed in the main pane.
  - g. Repeat step a through step f above. However, this time select the **test\_99.txt** file, and assign it the Feature ID **99**.
  - h. From the menu bar, select **Actions > Encrypt All**. The text files are protected.
4. Close Data File Protection utility and Envelope.

#### To access the protected data file:

1. Open each protected text file with Microsoft Notepad or with the unprotected version of the TextViewer application. Random characters are displayed.
2. Connect the Demo key to your machine.
3. Open the protected version of the **test\_42.txt** file with the protected version of the TextViewer application. The original text is successfully displayed. (The Demo key contains licenses for Features 0 and 42. Therefore, the protected TextViewer application can operate and the file that was protected with Feature 42 can be accessed.)
4. Open the protected version of the **test\_99.txt** file with the protected version of the TextViewer application. Random characters are displayed. (The Demo key does not contain a license for Feature 99.)

## Working With the dfcrypt Command Line Utility

The **dfcrypt** utility provides an alternative to the Sentinel LDK Data Protection utility. **dfcrypt** enables you to encrypt data files by specifying the relevant information in a command line instead of a graphical user interface.

The utility supports **Version 1** and **Version 2 data protection modes** as follows:

- > **For Version 1: dfcrypt** can be executed only on a Windows machine. To run the utility, you must connect a Vendor key or protection key with the relevant Vendor Code to your machine.
- > **For Version 2: dfcrypt** can be executed on a Windows or Linux machine. To run the utility, you must connect a Vendor key with the relevant Vendor Code to your machine.

The **dfcrypt** utility is located in the following path:

> For Windows:

```
%ProgramFiles(x86)%\Thales\Sentinel LDK\VendorTools\VendorSuite\
```

> For Linux:

```
\Linux\VendorTools\Envelope\
```

To encrypt or decrypt data files using **dfcrypt**, enter the following command:

```
dfcrypt <options> <source> <destination>
```

The parameters used in the **dfcrypt** command line are described in the table that follows.

Parameter	Description
<b>options</b>	List of options that indicate the function to be performed by the <b>dfcrypt</b> utility. See the tables of options that follow.
<b>source</b>	The file to be read and processed by the utility. To process multiple files, place the files in a directory and specify the name of the directory for this parameter.
<b>destination</b>	The file to be generated by the utility. If you specified a directory as the source, specify the name of the directory to contain the generated output.

The options that determine the function to be performed by the **dfcrypt** utility are described in the tables that follows. All options for a given execution of **dfcrypt** must be selected from the same table.

## Options for **Version 1** Data Protection Mode (Windows)

Option	Action
<b>-v:1</b> <b>--encver:1</b>	This specifies that the data protection mode is <b>Version 1</b> .
<b>-e</b> <b>--encrypt</b>	<b>dfcrypt</b> reads the source file or directory and generates an encrypted file or a directory of encrypted files. (This is the default action.)
<b>-d</b> <b>--decrypt</b>	<b>dfcrypt</b> reads an encrypted source file or directory and generates an unencrypted file or a directory of unencrypted files.
<b>-c:&lt;file&gt;</b> <b>--vcf:&lt;file&gt;</b>	Name of a Vendor Code file (mandatory).

Option	Action
<b>-k:&lt;key&gt;</b> <b>--key:&lt;key&gt;</b>	The encryption key to be used to encrypt or decrypt data files (mandatory). You must also specify this encryption key in Sentinel LDK Envelope for each protected application that will access the protected data files. The key may contain 1-8 printable characters. If you include special characters, enclose the entire command in quotation marks. For example: " <b>k:qe4&lt;!r^B</b> "
<b>-o</b> <b>--overwrite</b>	Overwrite destination files, if any.
<b>-r</b> <b>--recursive</b>	Enables recursive handling of all files in all subdirectories contained in the specified source directory.
<b>-q</b> <b>--quiet</b>	Suppresses output by excluding copyright information and the progress indicator. Only error messages are displayed. This is particularly useful in Makefile integration.

For example:

```
dfcrypt -v:1 -c:demoma.hvc -k:4873Asdb data.txt data_crypt.txt
```

Encrypts the file **data.txt** using the specified Vendor Codes file and encryption key. The encrypted file is written to **data\_crypt.txt**.

```
dfcrypt --encver:1 --decrypt --recursive --vcf:demomb.hvc --key:4873Asdb myInputs myOutputs
```

Decrypts all the files in the directory **myInputs** and in all contained subdirectories, using the specified Vendor Codes file and encryption key. The decrypted files are written to the directory **myOutputs**.

## Options for **Version 2** Data Protection Mode (Windows)

Option	Action
<b>-v:2</b> <b>--enver:2</b>	This specifies that the data protection mode is <b>Version 2</b> .
<b>-e</b> <b>--encrypt</b>	<b>dfcrypt</b> reads the source file or directory and generates an encrypted file or a directory of encrypted files. (This is the only available action. The decrypt action is not available for Version 2.)
<b>-c:&lt;file&gt;</b> <b>--vcf:&lt;file&gt;</b>	Name of a Vendor Code file (mandatory).

Option	Action
<b>-k:&lt;key&gt;</b> <b>--key:&lt;key&gt;</b>	The encryption key to be used to encrypt data files (optional). The key may contain 1-8 printable characters. If you include special characters, enclose the entire command in quotation marks. For example: " <b>-k:qe4&lt;!r^B</b> "  If you do not provide a key, each file will be encrypted using a random key.
<b>-f:&lt;fid&gt;</b> <b>--fid:&lt;fid&gt;</b>	The Feature ID to be used to license the data file. (The default Feature ID is 0.)
<b>-o</b> <b>--overwrite</b>	Overwrite destination files, if any.
<b>-r</b> <b>--recursive</b>	Enables recursive handling of all files in all subdirectories contained in the specified source directory.
<b>-q</b> <b>--quiet</b>	Suppresses output by excluding copyright information and the progress indicator. Only error messages are displayed. This is particularly useful in Makefile integration.

For example:

```
dfcrypt -v:2 -c:demoma.hvc -f:25 data.txt data_crypt.txt
```

Encrypts the file **data.txt** using the specified Vendor Codes file. Feature ID 25 is used to license the file. The encrypted file is written to **data\_crypt.txt**.

```
dfcrypt --encver:2 --recursive --vcf:demomb.hvc --fid:50 myInputs myOutputs
```

Encrypts all the files in the directory **myInputs** and in all contained subdirectories, using the specified Vendor Codes file. Feature ID 50 is used to license all the files. The encrypted files are written to the directory **myOutputs**.

## Options for **Version 2** Data Protection Mode (Linux)

Option	Action
<b>-c:&lt;file&gt;</b> <b>--vcf:&lt;file&gt;</b>	Name of a Vendor Code file (mandatory).
<b>-k:&lt;key&gt;</b> <b>--key:&lt;key&gt;</b>	The encryption key to be used to encrypt data files (optional). The key may contain 1-8 printable characters. If you include special characters, enclose the entire command in quotation marks. For example: " <b>-k:qe4&lt;!r^B</b> "  If you do not provide a key, each file will be encrypted using a random key.

Option	Action
<b>-f:&lt;fid&gt;</b> <b>--fid:&lt;fid&gt;</b>	The Feature ID to be used to license the data file. (The default Feature ID is 0.)
<b>-o</b> <b>--overwrite</b>	Overwrite destination files, if any.
<b>-r</b> <b>--recursive</b>	Enables recursive handling of all files in all subdirectories contained in the specified source directory.
<b>-q</b> <b>--quiet</b>	Suppresses output by excluding copyright information and the progress indicator. Only error messages are displayed. This is particularly useful in Makefile integration.

For example:

```
dfcrypt -c:demoma.hvc --key:1c03m06k -f:25 data.txt data_crypt
```

Encrypts the file **data.txt** using the specified Vendor Codes file and the specified encryption key. Feature ID 25 is used to license the file. The encrypted file is written to **data\_crypt**.

```
dfcrypt --recursive --vcf:demomb.hvc --fid:50 myInputs myOutputs
```

Encrypts all the files in the directory **myInputs** and in all contained subdirectories, using the specified Vendor Codes file. A random encryption key is used. Feature ID 50 is used to license all the files. The encrypted files are written to the directory **myOutputs**.

## Source or Destination Files for Linux

If a conflict exists between an option name and the source or destination file name, append the prefix **--<space>** to the file name to differentiate it from the option. For example, **--key** is one of the option names. To specify a source or destination file named **--key**, specify it as: **-- --key**

## Display the dfcrypt help screen

Specify one of the following to display the options available for the **dfcrypt** utility:

```
dfcrypt -h
```

```
dfcrypt --help
```

## Part 3 - Licensing

### In this section:

---

- > ["Introduction to Sentinel LDK-EMS" on page 90](#) – Provides an overview of Sentinel LDK-EMS and the major processes it facilitates, lists its prerequisites, and explains how to use the application.
- > ["Preparing Your Sentinel LDK Licensing Plan" on page 98](#) – Outlines the importance of licensing your software products, describes the licensing options provided by Sentinel LDK, and explains how to prepare a licensing plan for use with Sentinel LDK-EMS.
- > ["Implementing Your Sentinel LDK Licensing Plan" on page 108](#) – Describes how to use Sentinel LDK-EMS to define and manage the Features and Products included in your Sentinel LDK licensing plan, and how to maintain Products and licenses as circumstances change.
- > ["Sentinel LDK Entitlements, Production, and Development Tasks" on page 122](#) – Describes how to use Sentinel LDK-EMS to manage and produce entitlements, and to perform additional development-related tasks.
- > ["Sentinel LDK Administration and Customer Services" on page 142](#) – Describes how to use Sentinel LDK-EMS to define Sentinel LDK user details, maintain Batch Codes, configure system settings, perform manual Product activation and maintain customer data.
- > ["Sentinel Remote Update System \(RUS\)" on page 147](#) – Describes the Sentinel Remote Update System utility (RUS utility) and explains how to use the RUS utility to remotely update license data in deployed Sentinel protection keys.
- > ["Generating Sentinel LDK Reports" on page 153](#) – Provides an overview of the Sentinel LDK-EMS Reporting facility and describes some of the main features of the facility.
- > ["Cloud Licensing Using Sentinel LDK Cloud Portal" on page 156](#) – Provides an overview of cloud licensing and describes how to implement cloud licensing for your applications when using Sentinel LDK Cloud Portal to manage licensed users.



- 
- > ["Cloud Licensing Using Sentinel Admin Control Center" on page 171](#) – Provides an overview of cloud licensing and describes how to implement cloud licensing for your applications when using Sentinel Admin Control Center to manage licensed users.

# CHAPTER 8: Introduction to Sentinel LDK-EMS

This section provides an overview of Sentinel LDK-EMS and the major processes it facilitates. It also describes the user roles and their functions in Sentinel LDK-EMS, lists its prerequisites, and explains how to start using the application.

An alternative to Sentinel LDK-EMS, the Sentinel License Generation API, is also described.

*In this section:*

- > ["Sentinel LDK-EMS Overview" below](#)
- > ["User Types and User Roles in Sentinel LDK-EMS" on page 92](#)
- > ["Getting Started With Sentinel LDK-EMS" on page 95](#)
- > ["Sentinel License Generation API" on page 96](#)

**NOTE** This section provides high-level information on Sentinel LDK-EMS processes. For detailed practical instructions for using each function in Sentinel LDK-EMS, see the [Sentinel LDK-EMS User Guide](#).

## Sentinel LDK-EMS Overview

Sentinel LDK-EMS is a powerful role-based application designed to manage the business activities required to implement and maintain Sentinel LDK in your organization.

Sentinel LDK-EMS streamlines the major workflows in the licensing lifecycle of a protected software application, from the moment it is developed, through its packaging, marketing, selling, and order-taking, to its distribution and upgrading.

Sentinel LDK separates the software protection process (implemented with Sentinel Licensing API or Sentinel LDK Envelope) from the licensing and production processes (implemented with Sentinel LDK-EMS), enabling you to modify your company's licensing strategy as necessary when circumstances change, and to implement these changes quickly and efficiently.

## Sentinel LDK-EMS Major Workflows

Sentinel LDK-EMS is installed as a service under Windows. The Sentinel LDK-EMS Service handles three major workflows: license planning, order processing and production, and software activation.

## License Planning

Before starting to use Sentinel LDK-EMS, it is recommended that business decision-makers in your organization, such as product or marketing managers, prepare a licensing plan based on the company's licensing strategy.

The licensing plan identifies each individual functional component in your software applications that can be independently controlled by a license. In Sentinel LDK, these components are referred to as *Features*. A Feature may be an entire application, a module, or a specific functionality such as Print, Save or Draw. Over 64,000 Features can be defined using Sentinel LDK-EMS.

In addition, the licensing plan can include the *Products* that your company wants to sell and/or distribute for evaluation. In Sentinel LDK, a Product is a collection of one or more licensed Features that can be sold or distributed as an item.

After completing the licensing plan, the Features and Products can be defined in Sentinel LDK-EMS. The output of this process is a repository of Products that are stored in the Sentinel LDK-EMS database—ready for customer orders.

**NOTE** You can make changes to your licensing plan and license models at any time, adding Features and Products as required.

For additional information on preparing a licensing plan for use with Sentinel LDK, see ["Preparing Your Sentinel LDK Licensing Plan" on page 98](#).

For a description of the many types of licensing models you can implement using Sentinel LDK, see ["Part 5 - Licensing Business Models" on page 239](#).

For additional information on defining Features and Products in Sentinel LDK-EMS, see ["Implementing Your Sentinel LDK Licensing Plan" on page 108](#).

## Order Processing and Production

Staff in your organization's orders department receive and fulfill entitlements. An *entitlement* is an order for Sentinel LDK items, and can be one of the following:

- > An order for Products to be supplied with one or more Sentinel protection keys
- > A *Protection Key Update* that specifies changes to be made to the license terms and/or data stored in Sentinel protection keys that have already been deployed

Order processing personnel process the entitlement details using Sentinel LDK-EMS. The license terms of each Feature in the ordered Products may be specified when the Product is defined, or when the entitlement is processed.

When all the details of an entitlement have been defined, the entitlement can be produced. The Product details, including the license terms and memory data, are stored in the specified Sentinel protection keys at the production stage or when the Product is activated, and can be updated after the keys have been deployed.

For additional information on processing and producing entitlements in Sentinel LDK-EMS, see ["Sentinel LDK Entitlements, Production, and Development Tasks" on page 122](#).

## Software Activation and Online Updates

Product activation and online updates are performed using Sentinel LDK-EMS when your software is installed at the customer's site.

### Product Activation with Sentinel SL Keys

With Sentinel SL keys, the software is activated and usable only after the following steps are completed:

1. A Product Key is produced in Sentinel LDK-EMS and supplied to the end user.
2. Using the Customer Portal, the end user sends the Product Key to Sentinel LDK-EMS for validation.
3. A Sentinel SL key with license terms is sent back and installed on the end user's computer.

### Online Updates

Online updates can be implemented in the following ways:

- > The Protection Key Update information is stored in Sentinel LDK-EMS for use in software that you provide to your end users. The update is then implemented as part of the end users' installation process.
- > The Protection Key Update information is stored in Sentinel LDK-EMS. After the protection key is installed on the end user's machine, the user can access the Sentinel LDK-EMS Customer Portal. Sentinel LDK-EMS automatically applies any outstanding updates to the protection key. (See ["Customer Portal - Applying Updates to Protection Keys" on page 133](#).)
- > A file that contains the Protection Key Update information is generated and sent to the end user. This file can then be used with the Sentinel Remote Update System (RUS utility) utility or the Admin Control Center to ensure secure, remote updating of the deployed Sentinel protection keys.

For additional information on the RUS utility, see ["Sentinel Remote Update System \(RUS\)" on page 147](#).

A receipt can be generated when a Protection Key Update is processed, to verify that the update has been applied.

## User Types and User Roles in Sentinel LDK-EMS

---

Sentinel LDK-EMS is a role-based application. The functions and tasks that you can perform are determined by the user type and user roles assigned to you by the Sentinel LDK Administrator.

The following user types exist:

### > Vendor

This is a Sentinel LDK-EMS user within the software vendor's organization. The access rights and functionality that are available to each user are determined by the specific roles that are assigned to the user.

### > Channel Partner

This is a Sentinel LDK-EMS user in a channel partner's organization. This type of user is limited to a range of activities that relate to the specific channel partner.

**NOTE** The roles, access rights, and functionality described in the following sections are not relevant for users of the Sentinel LDK-EMS Customer Portal.

The roles, access rights and functionality for the Vendor and Channel Partner user types are described in greater detail in the sections that follow.

## Vendor

The table that follows describes the out-of-the-box roles that can be assigned to a vendor user. (Almost all of the tasks listed in the table relate to functionality in Sentinel LDK-EMS.)

Role	Authorized Tasks	Learn More
<b>Product Manager</b>	Define and manage Features and Products	<ul style="list-style-type: none"> <li>&gt; <a href="#">"Preparing Your Sentinel LDK Licensing Plan" on page 98</a></li> <li>&gt; <a href="#">"Implementing Your Sentinel LDK Licensing Plan" on page 108</a></li> </ul>
<b>Entitlement Manager</b>	Define and manage customers and channel partners Enter and manage entitlements	<a href="#">"Sentinel LDK Entitlements, Production, and Development Tasks" on page 122</a>
<b>Production</b>	Produce entitlements	<a href="#">"Sentinel LDK Entitlements, Production, and Development Tasks" on page 122</a>
<b>Customer Services</b>	Define and manage customers and channel partners Manage Product activations	<a href="#">"Sentinel LDK Administration and Customer Services" on page 142</a>
<b>Report Generation</b>	Run and view reports Schedule generation of and arrange distribution of reports.	<a href="#">"Generating Sentinel LDK Reports" on page 153</a>

Role	Authorized Tasks	Learn More
<b>Development</b>	Perform development-related tasks Operate Sentinel LDK ToolBox and Sentinel LDK Envelope	<a href="#">"Sentinel LDK Entitlements, Production, and Development Tasks" on page 122</a>
<b>Batch Code Admin</b>	Can perform the following functions for the assigned Batch Codes: <ul style="list-style-type: none"> <li>&gt; Manage Sentinel LDK users</li> <li>&gt; Maintain Sentinel LDK Master license</li> <li>&gt; Configure system settings</li> <li>&gt; Generate reports</li> <li>&gt; Manage scheduled reports</li> </ul>	<a href="#">"Sentinel LDK Administration and Customer Services" on page 142</a>
<b>Super User</b>	Can perform the following functions for all Batch Codes: <ul style="list-style-type: none"> <li>&gt; Manage Sentinel LDK users</li> <li>&gt; Maintain Sentinel LDK Master license</li> <li>&gt; Configure system settings</li> <li>&gt; Generate reports</li> <li>&gt; Manage scheduled reports</li> </ul>	<a href="#">"Sentinel LDK Administration and Customer Services" on page 142</a>

The “admin” user is authorized to perform all functions in Sentinel LDK. Only the admin user can assign the **Super User** role to another user.

## Channel Partner User

A Sentinel LDK-EMS user that is associated with a specific channel partner is referred to as a *Channel Partner user*.

A Channel Partner user can perform the following functions for the relevant channel partner customers:

- > Create and manage end-user customers.
- > View, produce, and activate entitlements.
- > Resend emails for entitlements.

- > Display product keys for entitlements.
- > Check in, browse, and view details of C2V files.

All other Sentinel LDK-EMS functionality is blocked for this type of user. When a Channel Partner user logs in to Sentinel LDK-EMS, the landing page is the Entitlements page.

To designate a user as a Channel Partner user, you must first obtain the **Channel Partner** module for your Sentinel LDK Master license. For more information, see ["Channel Partner Module" on page 317](#).

For more information on channel partner functionality in Sentinel LDK-EMS, see ["Channel Partners" on page 145](#).

## Getting Started With Sentinel LDK-EMS

Before you start to use Sentinel LDK-EMS, ensure that:

- > You have a URL to access Sentinel LDK-EMS.
- > You have received a Sentinel LDK-EMS user name and password from your Sentinel LDK-EMS system administrator.

After you have logged in to Sentinel LDK-EMS, change the Sentinel LDK-EMS password that you received to a password of your own choice. For additional information on changing your password, see the [Sentinel LDK-EMS User Guide](#).

**NOTE** Sentinel LDK-EMS passwords are case-sensitive, so ensure that you enter upper-case and lower-case letters correctly when you type your password.

### Prerequisites for the Sentinel LDK Administrator

If you are performing administration functions for Sentinel LDK in your organization, it is essential that you check the following requirements before you (or other users) start to use Sentinel LDK-EMS:

- > A valid connection to Sentinel LDK-EMS must exist. For additional information on installing and configuring Sentinel LDK-EMS, see the [Sentinel LDK Installation Guide](#).
- > One of your Vendor keys must be introduced to Sentinel LDK using the Master Wizard. For additional information, see the [Sentinel LDK Installation Guide](#).
- > **For Sentinel LDK-EMS on-premises:** The Master key must remain connected to the Sentinel LDK-EMS Server machine in order to enable you to perform Sentinel LDK-EMS functions. If Sentinel LDK-EMS is installed on more than one machine, each machine must have a separate Master key connected locally.

**NOTE** If you are evaluating Sentinel LDK-EMS installed on-premises, you can use the provided **DEMOMA** Batch Code, which does not require a Master key.

- > You must define user names, passwords, roles, and Batch Code access for each Sentinel LDK-EMS user, and also for yourself. For additional information, see ["Maintaining User Details" on page 143](#).

For additional information on the Sentinel LDK administration tasks and options in Sentinel LDK-EMS, see ["Administration Tasks" on page 142](#).

## Using the Sentinel LDK-EMS Help

Detailed instructions for using each function and task in Sentinel LDK-EMS are provided in the Sentinel LDK-EMS help.

To access the help for a specific screen, click the **Help** link in the top-right corner. Many individual screens also contain a help button for information about the contents of the screen.

You can also access the online Help—[Sentinel LDK-EMS User Guide](#).

## Sentinel License Generation API

For sites that already have a licensing infrastructure in place or that prefer to implement an alternative to Sentinel LDK-EMS, Sentinel LDK offers a standalone licensing solution.

You can use Sentinel License Generation API together with your existing licensing server software and ERP and CRM back office systems for maximum flexibility and control over your business processes.

Sentinel License Generation API provides the functionality required to generate and maintain Sentinel protection keys, but without any of the back office services that are provided by Sentinel LDK-EMS. All the required services are provided by the system that you choose to implement. You would use Sentinel LDK only to handle the protection and Feature-control functions for your applications.

Sentinel License Generation API is included in Sentinel LDK ToolBox. Documentation for the API is included in the ToolBox help system and is provided in the [Sentinel License Generation API Reference](#).

**NOTE** To generate licenses, the Master key must be connected to the machine where the program that calls Sentinel License Generation API is running. To connect the Master key from a remote machine, refer to the [Sentinel LDK Installation Guide](#).

## Switching Between Back-ends to Maintain Protection Keys

Sentinel LDK-EMS retains an update counter for each protection key that was created or updated using Sentinel LDK-EMS. This update counter must remain synchronized with the update counter that is stored in the protection key. If the synchronization is lost or was not set up correctly, you cannot use Sentinel LDK-EMS to update the existing Products in the protection key. A synchronization problem can result from various situations, including the following:

- > You used Sentinel License Generation API to update a protection key that had been created or updated using Sentinel LDK-EMS.



- > You attempted to use two (or more) installations of Sentinel LDK-EMS, each with its own database, to update a single protection key.
- > You used Business Studio (a legacy back-end) to update a protection key that had been created or updated using Sentinel LDK-EMS.

In these situations, some Products in the protection key may not be visible when you view the contents of the key in the Sentinel LDK-EMS Entitlements screen. Therefore, you cannot create an entitlement for these Products in Sentinel LDK-EMS.

**CAUTION!** Thales recommends that you not use multiple back-ends in parallel to update a given protection key. Using multiple back-ends may cause unexpected results.

If a synchronization problem occurs, obtain a C2V file for the protection key and check the C2V file into Sentinel LDK-EMS. This restores the synchronization between Sentinel LDK-EMS and the protection key.

**NOTE** New functionality in Sentinel LDK is often introduced first in Sentinel License Generation API and then, in a following release, in Sentinel LDK-EMS. Before you move from Sentinel License Generation API to Sentinel LDK-EMS, make sure that all the functionality that you are using is supported in Sentinel LDK-EMS. Otherwise, the C2V files sent by customers may contain parameters that Sentinel LDK-EMS does not recognize.

If you want, you can move from Sentinel LDK-EMS to Sentinel License Generation API because Sentinel License Generation API does not store the update counter. Instead, it relies on receiving the update counter each time in the C2V file sent by the customer.

# CHAPTER 9: Preparing Your Sentinel LDK Licensing Plan

Before you start to use Sentinel LDK-EMS in your organization, you may want to prepare a detailed licensing plan for use with Sentinel LDK. Although it is recommended that you prepare a licensing plan, it is not a prerequisite for using Sentinel LDK-EMS. Licensing decisions can be implemented or modified at any point.

This section outlines the importance of licensing your software products, describes the licensing options provided by Sentinel LDK, and suggests how you might prepare a detailed licensing plan for use with Sentinel LDK-EMS.

*In this section:*

- > ["Licensing Overview" below](#)
- > ["Preparing Your Licensing Plan" on the next page](#)
- > ["Choosing the Protection Level for Your Products" on page 101](#)
- > ["Designating Products for Trial or Grace Period Use" on page 104](#)
- > ["Assigning License Terms to Features" on page 104](#)
- > ["Utilizing Protection Key Memory" on page 106](#)
- > ["Using Your Licensing Plan With Sentinel LDK-EMS" on page 107](#)

**NOTE** This section provides high-level information about Sentinel LDK licensing options. For detailed practical instructions for implementing the licensing options in Sentinel LDK-EMS, see the [Sentinel LDK-EMS User Guide](#).

## Licensing Overview

["Part 2 - Protection" on page 36](#) in this guide explained in detail how to protect your software and intellectual property. In addition to protecting these valuable assets, it is essential that you protect your company's revenue by ensuring that your software is available only to the appropriate users, according to the terms that you define. This process is controlled by *licensing*.

Licensing provides you with the flexibility to implement your business strategies for the sale and distribution of your software products. You define the licensing terms with which your software is distributed or sold according to your decisions about what is commercially beneficial to your company.

For example, you may decide that you initially want to distribute your software free of charge, so that users can try it before purchasing. You will want to ensure that users can use it for only a limited time before it must be purchased.

Alternatively, you may publish very complex, expensive software. You may decide to make specific components of that software available for a lower price, thus making parts of it accessible to users who cannot afford the full-featured version.

The versatility of Sentinel LDK enables you to implement a wide variety of licensing models. For more information on the many models you can apply to your software offering, see ["Part 5 - Licensing Business Models" on page 239](#).

## Preparing Your Licensing Plan

---

A useful step in the development of a licensing strategy is the preparation of a *licensing plan*. Business decision-makers in your organization, such as product managers or marketing managers, define protection and business rules, and specify the licensing business models required to meet your company's business needs.

A *licensing business model* is the logic behind a business decision relating to the way a Product is licensed. For example, a rental licensing business model enables you to charge for the use of software for a specific period of time.

Sentinel LDK enables you to choose from a variety of out-of-the-box licensing business models, including:

- > Trialware (try-before-you-buy)
- > Rental/Subscription
- > Module-based
- > Feature-based
- > Floating users
- > Time-based
- > Execution-based
- > Perpetual
- > Unlocked

You can define additional licensing business models and software usage terms to meet your company's individual requirements.

It is recommended that you prepare a licensing plan before you start to use Sentinel LDK to streamline the implementation of your company's licensing strategy. Your Sentinel LDK licensing plan should be based on the detailed licensing requirements that you define for all the protected software applications to be sold by your company, and/or distributed for trial use.

The process of preparing a Sentinel LDK licensing plan can include the following steps:

1. Analyzing all the relevant software applications and identifying each functional component that can be licensed individually.
2. Combining these components into licensed entities that can be offered to customers.
3. Deciding which Sentinel protection keys you want to supply with your software applications.
4. Specifying the detailed licensing terms to be applied, according to your licensing strategy.

The output of such a process is a comprehensive licensing plan that can be implemented using Sentinel LDK-EMS.

**NOTE** You can make changes to your licensing plan and licensing business models at any time.

## Identifying Functional Components (Features)

The recommended first step in evaluating and planning your licensing requirements involves analyzing your software applications and identifying their functional components. Most applications can be segmented into a number of distinct functional components. In Sentinel LDK, these components are referred to as *Features*.

Each individual Feature is an identifiable functionality of a software application that can be independently controlled by a license. In Sentinel LDK, a Feature may be an entire application, a module or a specific functionality such as Print, Save or Draw.

### Example: Specifying Features

**Scenario:** The Product Manager of High Quality Software Ltd. (HQ Software), a company providing design software for the construction industry, identifies the specific functional components that the company wants to license, and assigns a Feature name to each component.

The following table lists the defined functional components and the Feature names assigned to each component:

Functional Component	Feature
Drawing design plans	DRAW
Viewing design plans	VIEW
Saving projects	SAVE
Printing designs	PRINT DESIGNS
Printing predefined reports	PRINT REPORTS
Generating tailored reports	REPORT GENERATOR

## Combining Features Into Products

After you have identified and listed all the individual Features to license, you can define the different combinations of licensed Features that your company wants to sell.

In Sentinel LDK, a collection of one or more licensed Features that can be sold as an item is referred to as a *Product*. Products can differ from each other, not just in the Features that they contain, but also in the license terms specified for each Feature.

Your licensing plan can contain the names of all the Products that your company wants to sell and/or distribute for evaluation, and the Features that each Product includes.

In Sentinel LDK, you have full control over the specific Products you define, the Features they include, and the license terms assigned to each Feature in each Product.

## Example: Defining Products

**Scenario:** The HQ Software Product Manager decides to define a trial Product intended for distribution to customers who want to evaluate their software. This Product, **HQ Design Demo**, includes only the VIEW and PRINT DESIGNS Features.

In addition, the company defines:

- > A Product intended for small-office customers, **HQ Design Lite**, offering the Features included in **HQ Design Demo**, with the addition of DRAW and SAVE
- > A Product targeted towards larger customers, **HQ Design Pro**, that offers all available Features

In this scenario, the REPORT GENERATOR Feature is not yet fully developed and is not currently included in the **HQ Design Pro** Product.

## Choosing the Protection Level for Your Products

---

Your choice of the Sentinel protection keys to be distributed together with your licensed software reflects the level of protection you wish to apply and the way you intend to control the use of or access to each Product.

Two types of Sentinel protection keys are available:

- > **Sentinel HL keys:** The hardware-based protection and licensing component of Sentinel LDK that provides the safest and strongest level of protection.
- > **Sentinel SL keys:** The software-based protection and licensing component of Sentinel LDK—virtual Sentinel HL keys. Sentinel SL keys are further divided into **AdminMode** and **UserMode** keys

For more information on the different types of keys and a comparison of the benefits for each type, see ["Sentinel HL Keys" on page 26](#).

Your software and the user license are both locked to the Sentinel protection key that you select. When you define the Product licenses to be included in your licensing plan, you also select which Sentinel LDK *locking type* to assign to each Product license. The locking type that you select determines the level of protection for each Product license as follows:

Locking Type	Level of Protection Provided
HL	Hardware-based level of protection.
SL AdminMode	Software-based level of protection.
SL UserMode	
HL or SL AdminMode	Hardware- or Software-based level of protection. <ul style="list-style-type: none"> <li>&gt; If a Product is shipped solely with Sentinel HL keys, then a hardware-based level of protection is achieved.</li> </ul>
HL or SL (AdminMode or UserMode)	<ul style="list-style-type: none"> <li>&gt; If a Product is shipped even once with Sentinel SL keys, then the overall level of security should be considered to be that provided by SL keys. This is because there is always the possibility that an attacker could have access to a deployed SL key.</li> </ul>

**When installing an SL license:** If the locking type that is specified for a Product license is **HL or SL (AdminMode or UserMode)** and the Run-time Environment is present on the user's machine, then **SL AdminMode** is selected automatically. Otherwise, **SL User Mode** is installed.

## Sentinel HL Key Activation

A Product that is protected with a Sentinel HL key can be activated only after the end user receives a Sentinel HL key containing the license terms for the Product and connects the key to the computer.

For details on burning the Sentinel HL key, see [Sentinel LDK–EMS User Guide](#).

For additional information on the usage and benefits of Sentinel HL keys, see ["Sentinel HL Keys" on page 26](#).

## Sentinel SL Key Activation

A Product that is protected with a Sentinel SL key can be activated using one of the methods that follow.

For additional information on the usage and benefits of Sentinel SL keys, see ["Sentinel HL Keys" on page 26](#).

## Activation Using a Product Key

1. A *Product Key*, consisting of a string of characters, is generated in Sentinel LDK-EMS and supplied to the end user. This can be done by including the Product key in the physical installation package or by providing the Product key by email as part of the entitlement process.
2. The end user returns the Product Key as proof of purchase.
3. The Product Key is sent to Sentinel LDK-EMS for verification.

4. A Sentinel SL key with license terms is sent back and installed on the end user's computer.

(The end user can perform steps 2, 3, and 4 automatically with the Sentinel LDK-EMS Customer Portal.)

**NOTE** If the locking type that is specified for a Product license is **HL or SL (AdminMode or UserMode)** and the Run-time Environment is present on the user's machine, then **SL AdminMode** is selected automatically. Otherwise, **SL User Mode** is installed.

## Activation Using a Protection Key Update

The steps that follows can be performed:

- > Manually using Admin Control Center or the RUS utility.
  - > Programmatically using Sentinel Licensing API or Sentinel Web Services.
1. A fingerprint of the end user's machine is generated in a C2V file.
  2. The C2V file is sent to the vendor (typically by email).
  3. The vendor uses the C2V file to generate a Product activation update (in a V2C file) using Sentinel LDK-EMS or using Sentinel License Generation API.
  4. The V2C file is sent to the end user.
  5. The V2C file is used on the end user's machine to install an SL key.

## Specifying the Protection Level for Individual Orders

Sentinel LDK gives you the flexibility to choose the Sentinel protection keys for a Product or according to the requirements of each individual order.

If you prefer not to specify the protection level in advance, you can assign the **HL or SL AdminMode or SL UserMode** locking type to a Product. With this locking type, the decision on which type of Sentinel protection key is to be shipped with the Product is made when each order is processed.

**NOTE** Although SL keys provide a high level of protection, HL key security is superior. A Product whose locking type allows for both HL and SL keys provides HL key-level protection if the Product is only shipped with HL keys (that is, the Product is *never* shipped with SL keys). *However if the Product is also sometimes shipped with SL keys, the overall level of security should be considered to be that provided by SL keys.* This is because there is always the possibility that an attacker could have access to a deployed SL key.

## Designating Products for Trial or Grace Period Use

Sentinel LDK enables you to create, protect, and distribute secure *trialware* versions of your software. You can invite users to download your trial software from networks, to share it with other users, and to give it away to their friends or colleagues. End users then have the option to purchase your software and to turn their trial copy into a fully-licensed version by activating it with a Sentinel protection key.

You can also use Sentinel LDK to define *grace periods* for your software. During the grace period, and even after activation, end users can pass copies of their purchased software to as many friends as they wish. When a friend installs the software, it automatically reverts to a limited trial version for the entire grace period. After the grace period expires, the software can no longer run until it is activated with a Sentinel protection key.

Sentinel LDK enables you to define trial and grace periods of up to 90 days for software protected with any type of Sentinel protection key.

For example, software protected with Sentinel HL keys can be purchased and delivered over the Internet with a built-in grace period while the Sentinel HL keys are shipped. The end users can start using the software immediately while waiting for the arrival of their Sentinel HLkeys.

Similarly, end users who purchase and install a software application can use it for a grace period without activating it. During this grace period, they can activate the software remotely and receive a Sentinel SL key, after which the software will run according to the purchased license terms stored in the keys. If the grace period expires and the software has not been activated, the software stops running until activated by the end user.

In Sentinel LDK, a Product that is intended for distribution as trialware or for use during a grace period is referred to as an *unlocked trialware Product*.

Your licensing plan can include all the unlocked trialware Products to be offered by your organization.

**NOTE** You may be able to issue an unlocked Product for any length of time. This depends on the type of module that you subscribe to for your [Sentinel LDK Master license](#). For details, see ["Defining Unlocked Products" on page 115](#).

## Assigning License Terms to Features

Sentinel LDK enables you to assign individual *license terms* to each Feature in each Product that you define. You can also define Products that include the same Features, but with different license terms. Such decisions are based on the commercial requirements of your organization, and on the license models that you choose to implement.

You can control Feature usage through the license by specifying the *license type* to be applied. You can choose one of the following license types:

- > **Perpetual:** Indicates that the Feature can be used an unlimited number of times for an unlimited period of time.



- > **Time Period:** Indicates that the license for the Feature expires after a specific number of days, starting from the date of first use.
- > **Expiration Date:** Indicates that the license for the Feature expires on a specific date.

When using Sentinel License Generation API, you can optionally define a start date for the license.

For example: If you want to provide a customer with a 30-day license that expires on a specific date, you can deliver the license any time prior to the start date specified in the license. The customer will be able to use the license only from the specified start date.

**NOTE** Expiration date licenses can be used starting from 00:00:00 UTC on the start date and expire at 23:59:59 UTC on the expiration date. You can optionally specify different times in the license definition.

If no start date is specified, the license is active as soon as it is received and installed by the customer.

The following limitations apply for the use of start date:

- Only applicable for SL AdminMode and SL UserMode keys.
  - Requires Sentinel Run-time Environment version 9.12 or later.
- > **Execution Count:** Indicates that the license for the Feature expires after a specific number of product executions.

After you select the type of license to apply to each Feature in a Product, you can specify its value, for example, the number of times that a Feature can be used.

If the Feature is intended to be used on a network or remote desktop, you can also specify the number of concurrent instances (*network seats*) allowed, and you can specify how concurrent instances are to be counted for the purpose of the license. (Basic keys do not support concurrency or remote desktops.) In addition, if the Feature will be used in Products that are locked to Sentinel SL keys, you can specify that the Feature and its license may be temporarily detached from the network for attachment to a remote recipient machine.

## Specifying License Values for Individual Orders

Sentinel LDK offers you maximum flexibility with regard to license terms, enabling you to supply the same Product to different customers with different license term values.

You do not have to specify in advance the exact values for the license, including the number of concurrent instances for each Feature in the Product. When each order for the Product is processed, the person processing the order can define the values required for that specific order.

## Example: Specifying License Terms and Protection Levels

**Scenario:** The HQ Software Product Manager decides to specify the following license terms for its three Products:

- > A trial period of 30 days for the PRINT and VIEW Features in its **HQ Design Demo** Product

- > A low-cost annual rental license for the DRAW and SAVE Features in the **HQ Design Lite** Product, with unlimited usage for the PRINT and VIEW Features
- > A more costly, full-featured license for the **HQ Design Pro** Product that specifies unlimited usage for all Features

The following protection levels are defined for each of the Products:

- > **HQ Design Demo** is defined as an unlocked trialware Product, to enable it to be distributed freely for evaluation
- > **HQ Design Lite** is supplied with Sentinel SL key protection, enabling electronic distribution
- > **HQ Design Pro** is supplied with Sentinel HL key protection, for maximum security

The following table summarizes the three Products, their protection levels, and their licensed Features:

Product:	HQ Design Demo	HQ Design Lite	HQ Design Pro
Protection Level:	Unlocked trialware	Sentinel SL keys	Sentinel HL keys
Licensing Business Model:	Trial	Rental	Unlimited
Feature			
DRAW	–	Expires after 1 year	Unlimited
VIEW	30 days	Unlimited	Unlimited
SAVE	–	Expires after 1 year	Unlimited
PRINT DESIGNS	30 days	Unlimited	Unlimited
PRINT REPORTS	–	–	Unlimited
REPORT GENERATOR	–	–	Not yet available

## Utilizing Protection Key Memory

All Sentinel protection keys—with the exception of Sentinel HL Basic keys—contain secure internal read-only and read/write memory. You can define specific segments for memory data and choose whether the data is added when you create a Product or when an order is being processed.

You can use memory data, for example, to:

- > Store licenses from your own licensing schemes
- > Save passwords, program code, program variables, and other data

Memory data can be defined for each Product. The contents of the memory are transferred to the secure memory of the selected Sentinel protection keys together with the Features, license terms and other data defined for the Product.

You can add any specific data that is required to be stored in memory for each Product to your licensing plan. For more information, see ["Defining Protection Key Memory Data" on page 111](#).

## Using Your Licensing Plan With Sentinel LDK-EMS

---

Your licensing plan can be implemented using Sentinel LDK-EMS. As your licensing requirements change, you can revise the licensing plan and ensure that the changes are implemented using Sentinel LDK-EMS. Your licensed Products can be easily and securely updated as required, after they have been deployed to customers.

For additional information on implementing and maintaining your licensing plan, see ["Implementing Your Sentinel LDK Licensing Plan" on page 108](#).

Sentinel LDK offers you the flexibility to update your licensing strategy as necessary, and to adapt rapidly to changes in the market, in your company's business strategy, or in customer purchasing preferences.

# CHAPTER 10: Implementing Your Sentinel LDK Licensing Plan

This section is intended for Sentinel LDK-EMS vendor users who are assigned the **Product Management** role. It describes how to use Sentinel LDK-EMS to define and manage Features and Products in Sentinel LDK, and to maintain Products and licenses as circumstances change.

For information on preparing a licensing plan and on Sentinel LDK licensing options, see ["Preparing Your Sentinel LDK Licensing Plan" on page 98](#).

For an overview of Sentinel LDK-EMS and for information on starting to use the application, see ["Introduction to Sentinel LDK-EMS" on page 90](#).

*In this section:*

- > ["License Planning in Sentinel LDK-EMS" below](#)
- > ["Managing Features" on the next page](#)
- > ["Managing Products" on page 110](#)
- > ["Maintaining Products and Licenses" on page 118](#)

**NOTE** This section provides high-level information on license planning and definition processes. For detailed practical instructions for using each function in the [Sentinel LDK-EMS User Guide](#).

## License Planning in Sentinel LDK-EMS

Before you start to use Sentinel LDK-EMS for license planning, it is suggested that you prepare a licensing plan. For additional information, see ["Preparing Your Sentinel LDK Licensing Plan" on page 98](#).

When you start Sentinel LDK-EMS, you have access to the Licensing Plan group of functions, including:

- > Managing Features
- > Managing Products

Each of these functions is described in this section.

**NOTE** All Sentinel LDK Features and Products are associated with a Sentinel LDK Batch Code. For additional information on Batch Code, see ["Personalized Vendor and Batch Codes" on page 21](#).

## Managing Features

You can perform the following tasks using the **Catalog > Features** tab in Sentinel LDK-EMS:

- > Define Features
- > Withdraw Features from use
- > View the details of all defined Features associated with the selected Batch Code

### Defining Features

If you have prepared a licensing plan, the first stage in its implementation is to use Sentinel LDK-EMS to define all the Features that you listed in the plan.

Before you begin to define Features, ensure that you have the following information available for each new Feature:

- > (Mandatory) The Batch Code associated with the Feature
- > (Mandatory) A Feature Name that is unique in the selected Batch Code.
- > (Mandatory) An ID number for the Feature. You can use the default Feature ID that Sentinel LDK-EMS assigns automatically for each new Feature, or you can assign your own numeric identifier. For example, you may want to maintain consistency with existing Feature data. The Feature ID that you specify must be unique in the selected Batch Code. The same Feature ID may be used in more than one Batch Code.
- > (Optional) A free-text description that provides additional information about the Feature, as well as other reference attributes.

After a Feature is included in one or more Products, the **Feature Name** and **Feature ID** cannot be changed. Other attributes, such as the description, can still be edited.

**NOTE** Although license terms are Feature-specific, they are not defined as part of the Feature. Instead, the license terms for a Feature are specified when the Feature is added to a Product, or when the Product is added to an entitlement. This is because the same Feature may be included in a number of Products, and the license terms for the Feature may vary according to the requirements of the Product or entitlement.

## Transferring Feature Definitions for Development Use

After you define the Features for a selected Batch Code, users authorized to perform Development tasks can transfer the Feature data to a file that can be used for development and protection purposes. For more information on transferring Feature definitions, see ["Exporting Definition Data" on page 137](#).

## Deleting Features

If the Feature has not been included in any Product, you can delete it. A Feature cannot be deleted once it has been deployed in at least one Product.

## Managing Products

You can perform the following tasks using the **Catalog > Products** tab in Sentinel LDK-EMS:

- > Define new Base Products
- > Define new Unlocked Products
- > Copy existing Products
- > Define new Modification Products
- > Define Cancellation Products
- > Open a Product to view or modify details
- > Withdraw Products from use
- > Restore Products that have been made obsolete
- > Delete a Product

**NOTE** You cannot modify license terms for a Product or delete a Product that has been fully defined (with the **Complete** status).

## Specifying the License Terms for Features in a Product

When you include a Feature in a Product, the following default license terms are assigned:

- > **License type:** Perpetual
- > **Number of concurrent instances:** Unlimited

To specify the required license terms for the Feature, you can:

- > Select a different license type:
  - **Expiration Date**
  - **Execution Count**

- **Time Period**

> Assign a value for the selected license type, for example:

- The expiration date
- The number of executions
- The number of days until the license expires, from the date of first use

If the Feature is intended to be used on a network, virtual machine, or remote desktop, you can specify the number of concurrent instances allowed, and you can select how concurrent instances are counted:

- > **Station:** All login requests from the same machine are counted as a single instance (default).
- > **Process:** All login requests from the same process are counted as a single instance.
- > **Login:** Each login request is counted as a different instance.

If the Feature is in a Product that will be locked to a Sentinel SL key, and is defined to be used on a network, you can specify that the license is allowed to be temporarily detached from the network pool. This means that the license can be attached to a remote recipient machine that is not connected to the network, to enable a user to work offline.

If required, you can specify that a user working in Remote Desktop (terminal machine) mode can access the license. Similarly, you can specify that the license for a Feature in a Product that will be locked to a Sentinel SL key can be enabled to run on a virtual machine.

You can leave the value for the license type undefined at this stage, and specify that the exact value will be defined when each order for the Product is processed.

Similarly, you can specify that the number of concurrent instances will be defined when an order for the Product is processed.

If you choose to make a Feature *excludable*, you enable the decision about whether the Feature is to be included in a specific order to be made at the time the order is being produced.

**NOTE** The above license term options do not apply to Unlocked Products. For additional information, see ["Defining Unlocked Products" on page 115](#).

## Defining Protection Key Memory Data

Memory data is used for storing strings in a Sentinel key. You might use memory data to migrate existing license data to Sentinel LDK-EMS, store login credentials for an application, or store homegrown license data, for example.

When you define a Product in Sentinel LDK, you can define the layout and contents of the memory data associated with the Product. At the customer site, memory data is stored in the protection key on the end user's computer or network.

**NOTE** You can create a Product with no Features whose only purpose is to define protection key memory. For more information, see ["Specifying the License Terms for Features in a Product" on page 110](#).

Sentinel LDK provides the following types of Protection Key memory:

### > Default memory

Default memory is available in all Sentinel HL and SL keys, except for Sentinel HL Basic keys. The amount of memory available depends on the type of key.

Default memory is divided into two fixed partitions:

- **Read/Write memory:** Data that can be updated when the deployed protected application is running, such as dynamic values for counters, or information retrieved during interaction with the user.
- **Read-Only memory:** Data that can be read when the protected application is running but cannot be changed. For example: the Product version number, text to be used in a "Welcome" message, fixed threshold values for counters.

You can divide each partition into multiple segments and enter data into them as required. Each segment is defined by an offset from the start of the partition and a length. It is the developer's responsibility to keep track of the location and size of each segment. You can redefine the data and the layout of each partition as required. In Sentinel LDK-EMS, you can specify that data is entered in one or more of the memory segments at order time.

### > Dynamic memory

Dynamic memory is available in:

- All Sentinel SL keys except for SL Legacy keys.
- All Sentinel HL (Driverless configuration) keys except for Sentinel HL Basic keys and Sentinel HL Pro keys. Dynamic memory is not available in Sentinel (HASP configuration) keys.

Dynamic memory is significantly larger than Default memory. However, dynamic memory space is shared between dynamic memory files (the space available to you for your applications, similar to default memory files) and license data (Features and Product). All space that is not utilized for license data can be used for dynamic memory files. For more information, see ["Maximum Number of Features in a Sentinel HL Key" on page 322](#).

Dynamic memory can be divided into dynamic memory files. Each file is assigned an identifier, which is used by your application to refer to that file. You can assign a size to the file at the time you create it, or allow the file size to be assigned automatically based on the amount of data that is written to the file. The following types of dynamic memory files can be created:

- **Read/Write file:** Data that can be updated when the deployed protected application is running, such as dynamic values for counters, or information retrieved during interaction with the user.



- **Read-Only file:** Data that can be read when the protected application is running but cannot be changed. For example: the Product version number, text to be used in a “Welcome” message, fixed threshold values for counters.
- **Read/Write-Once file:** Data that can be updated once when the deployed protected application is running. After a successful update, the memory becomes read-only memory.

You can set up Protection Key memory using either Sentinel LDK-EMS or Sentinel License Generation API. You can use any of the different types of memory to store and control licenses from your own licensing schemes. For information on the amount of memory available for each type of Sentinel HL key, see the [Sentinel HL Data Sheet](#). Sentinel SL keys contain 2,048 bytes of read-only Default memory and 4,032 bytes of read/write Default memory.

**NOTE** The memory in the protection key is shared by all Products in the key. When you allocate memory for a Product: Make sure that the memory space does not conflict with memory space for any other Product that may be protected with the same protection key.

The data defined in memory is written to the secure memory of the Sentinel protection keys together with the Features, license terms and other data defined for the Product.

For additional information on the use of memory files in Sentinel LDK-EMS, see [Sentinel LDK-EMS User Guide](#).

## Protecting Against Cloning

This section describes the protection of your protected application against attempts to clone the physical or virtual machine on which the protected application is installed.

### About Clone Protection

One of the methods sometimes employed to enable the unauthorized use of licensed software is machine cloning. Machine cloning involves creating an image of one machine (including your software and its legitimate license) and copying this image to one or more other machines. If there is no way to detect that the new image is running on different hardware than that on which it was originally installed, multiple instances of the software are available even though only a single license was purchased.

Sentinel LDK can detect probable machine cloning and disable protected software that is locked to Sentinel SL keys. Clone detection is effective whether the protected software is installed on a physical machine or on a virtual machine.

**NOTE** Cloning is only an issue for Sentinel SL keys. When software is locked to a Sentinel HL key, the physical key must be present in order for the software to run. Even if a machine image, including your software, is cloned, the software cannot run without the Sentinel HL key to which the software license is locked.

Protection against cloning is applied automatically when a protected application is locked to a Sentinel SL key.

For each Feature, you specify whether you want to allow the Feature to be accessible on virtual machines at the time you add the Feature to the Product or when preparing the order for the Product. By default, each Feature is accessible on virtual machines.

The clone protection functionality is tuned to minimize the occurrence of potential false positives (detection of a clone when no cloning exists), and reduce unnecessary calls to your technical support. As a result, it is possible that the clone protection functionality may not detect a cloned machine in every case. However, the possibility of this occurrence is low, especially when physical machines are cloned.

**NOTE** It is assumed that a customer's IT department follows best practices to avoid the collisions that would result from cloned machines that have identical UUID, MAC addresses or hostnames. When software is locked to a Sentinel SL key, the clone protection provided by many of the virtual machine clone protection schemes is based on this premise.

If you are concerned that your customers may be willing to accept collisions in order to attempt to bypass clone protection, consider one of the other Sentinel LDK solutions that provides a different tradeoff of security and convenience and is not affected by such deployment. A remote license (SL AdminMode or Sentinel HL) will provide the higher level of security that you require.

When the Sentinel LDK Run-time Environment detects cloning, it disables the licenses for which clone protection was specified. The end user is unable to log in to the software for which cloned licenses have been detected. The end user must activate the software before it can be used. Other licenses for which clone protection was not specified are not affected and the user may continue to log in and use the applications.

Detection of cloned licenses is recorded in the Sentinel License Manager and displayed in the Sentinel Admin Control Center. For additional information, see the [Sentinel Admin Control Center help](#).

For licenses locked to Sentinel SL keys, you enable and manage clone detection at the following points in the Product life cycle:

#### > During software protection

During protection of your software, use the Sentinel Licensing API to define how your application should behave when machine cloning is detected. For example, the application might display a message telling the end user that the software is disabled due to clone detection and that they should contact your customer services team.

**NOTE** If you use only Sentinel LDK Envelope for applying protection, (that is, without incorporating any additional software engineering), software that is disabled due to detection of cloning will return the following message to the end user: **Unknown error. H64**

#### > During Product definition:

When defining Products in Sentinel LDK-EMS:

For each Feature, decide whether the Feature should be accessible on virtual machines (this can also be decided during order entry). By default, accessibility on virtual machines is enabled.

### > During Product activation:

When Sentinel LDK-EMS detects cloning via the C2V file, it disables the protected application on the end user's machine.

To enable the protected application on the end user's machine, the end user must send a new fingerprint for the machine. This fingerprint can be generated with the RUS utility, or with the **GetInfo** function in Sentinel Licensing API. Use the fingerprint to generate a new entitlement for the end user.

When you attempt to check in a C2V file, Sentinel LDK-EMS blocks the action if it detects that the C2V file is from a cloned machine. Similarly, you cannot use a C2V file from a cloned machine to create a license update.

You can click **View Details** in the Check in Key screen to view details of the C2V if required.

## Simplified Clone Protection

A clone protection scheme defines which factors are considered by the Sentinel License Manager in order to determine whether a given Sentinel SL key has been cloned. You select the clone protection scheme when you define the Product.

Sentinel LDK offers several different clone protection schemes to protect applications that execute on physical machines and on virtual machines. The schemes are designed to accommodate a variety of circumstances. For example, schemes are available for applications that run on PCs, on Android machines, or on Microsoft Azure virtualization platforms. New schemes are added periodically as environments are added and evolve.

Keeping up with the latest developments in clone protection schemes can be a burden for most vendors. In addition, newer schemes may require that you install more recent versions of the Sentinel LDK Run-time Environment (for SL AdminMode licenses) or API libraries (for SL UserMode licenses) on the end users' machines.

Sentinel LDK provides a mechanism to simplify the process of implementing the most appropriate clone protection scheme for each situation. When you define a Product in Sentinel LDK-EMS, you can specify a clone protection scheme called **Platform Default** instead of choosing a specific scheme. When the Product license is installed on the end user's machine, Sentinel LDK automatically selects the most appropriate clone protection scheme for the type of operating system and the environment in which the license will be installed. (A similar mechanism is available when using Sentinel LDK License Generation API.)

For advanced users, more information on the **Platform Default** scheme and other clone protection schemes is available at ["How Sentinel LDK Detects Machine Cloning" on page 323](#).

## Defining Unlocked Products

An Unlocked Product is a license that can be installed by any user on any number of machines. An Unlocked Product is bundled with an SL AdminMode key or SL UserMode key as described below. The Unlocked Product is installed together with the protected application. The bundled protection key is activated automatically the first time the protected application runs.

If an Unlocked Product license has expired on a given machine, the user cannot reinstall the license to continue using the protected application.

An Unlocked Product is used:

- > to distribute a protected application as trialware during a grace period (typically 30 to 90 days or 30 executions).
- > to distribute a protected application that does not require any licensing restrictions and that can be used for an extended period of time or perpetually. The application is only protected against disassembly (although it may have some time limitation). You can choose to use a licensing mechanism other than Sentinel LDK to license the application (or you can choose to impose no additional license restrictions on the application).

**NOTE** To generate an Unlocked Product, you must purchase or subscribe to the relevant modules for your Sentinel LDK Master license.

An Unlocked Product can be defined for any type of Base Product. The properties of an Unlocked Product are similar to those for a standard Product, with the following exceptions:

- > **Locking Type:** Unlocked Products are automatically assigned the **SL AdminMode** locking type. If the Base Product on which the Unlocked Product is based has the **SL UserMode** locking type, the Unlocked Product is assigned the **SL UserMode** locking type.
- > **License Terms:** The maximum duration or maximum number of executions that you can define for any Feature in an Unlocked Product depends on the modules you have purchased or subscribed to for your Sentinel LDK Master license. The table that follows indicates the maximum value that can be assigned for a Feature in an Unlocked Product.

Licensing Limitation	Modules in Sentinel LDK Master license	
	Unlocked Trialware	Unlocked Unlimited (With or Without Unlocked Trialware)
Maximum duration of <b>Time Period</b> license	90 days from first execution	3,650 days from first execution
Maximum duration of <b>Expiration Date</b> license	90 days from current date	31-Dec-2091
Maximum number of executions for <b>Execution Count</b> license	30	16,777,215
Availability of <b>Perpetual</b> license	Not available	Available

**NOTE** The use of the Execution Count license type for Unlocked Products is only supported when working with Sentinel License Generation API.

The ["Default Feature" on page 374](#) in an Unlocked Product is automatically assigned one of the following durations, depending on the modules in the Sentinel LDK Master license:

- > For Unlocked Trialware: 90 days from first execution
- > For Unlocked Unlimited (with or without Unlocked Trialware): Perpetual

For additional information on the purpose and use of Unlocked Products for trialware, see ["Designating Products for Trial or Grace Period Use" on page 104](#).

For additional information on modules for the Sentinel LDK Master license, see ["Understanding the Sentinel LDK Master License" on page 305](#).

Unlocked Products are not available for inclusion in customer orders. Users authorized to perform Development tasks can bundle Unlocked Products for distribution. For additional information, see ["Creating Bundles of Unlocked Products" on page 136](#).

## Product Status Values

A Product can be assigned one of the following statuses:

**Draft** - The Product is not ready for distribution. The Product can be modified or deleted.

**Complete** - The Product can be included in an entitlement. The Product can be modified. However, it cannot be deleted. You can change its status to **End of Life** if you do not want the Product to be distributed any longer. Once the Product has been included in an entitlement ("Deployed"), the license terms can no longer be modified.

**End of Life** - The Product cannot be included in an entitlement. The Product's license terms cannot be modified. However, if you edit and save the Product, its status changes back to **Complete** and the Product can again be included in an entitlement.

## Duplicating a Product

After you have defined a Product, you can define additional Products with similar details using the **Copy** option in Sentinel LDK-EMS. This option creates a new Product using the defined properties, Features, and memory contents of the original Product, and enables you to make any changes you require, with the exception of changing the Base Product or the Product locking type (for example, SL-AdminMode) and locking status (locked/unlocked).

**NOTE** If you create a copy of a Base Product, you can give it a new name.

## Withdrawing a Product

At some stage, you may want to withdraw a selected Product from use and specify that it can no longer be included in orders, for example, if it is being replaced by an updated version.

If the Product has the status **Draft**, you can delete it. A Product cannot be deleted once it has been assigned the status **Complete**. You can, however, withdraw the Product from use by marking it as **End of Life**.

A withdrawn Product cannot be added to entitlements, but its details are maintained in Sentinel LDK-EMS for tracking purposes, and it continues to be functional when already at the end user's site.

## Restoring a Product

A Product whose status is **End of Life** can be restored to the **Complete** status. A restored Product can be used in the same way as any other Product.

## Maintaining Products and Licenses

---

After you have defined the initial Features and Products, you can use the Licensing Plan options in Sentinel LDK-EMS to cater for changing circumstances, such as the release of new software versions and changes in customer requirements.

Sentinel LDK-EMS enables you to maintain your licensing plan by defining new Features and Products as required. In addition, you can use Sentinel LDK-EMS to:

- > Manage Product versions
- > Cancel Product licenses

## Managing Product Versions

After you have implemented your initial licensing plan, you need to continue to review and update it to allow for changes in your company's software applications, in customer demand, in the market, and other considerations. For example:

- > Your company develops an enhanced version of an existing Product and you want to offer the new versions for sale instead of (or in addition to) the original Products.
- > You want to offer your existing customers the opportunity to replace their current version of a Product with an upgraded version that has additional Features.
- > Feedback from your customers indicates that they want to purchase a specific Product with different license terms than you are currently offering.

In circumstances such as these, since you cannot change the properties of an existing Product after it has been ordered, you can define a Modification Product based on the Base Product.

A *Modification Product* is a modified version of an existing Product, containing changes such as:

- > A software upgrade
- > Extended license terms
- > Added or removed Features

You can define several Modification Products for the same Base Product, with different Features, memory and/or license terms.

**NOTE** You can also define Modification Products based on an existing Modification Product.

## Defining a Modification Product

Before you start to define a Modification Product, ensure that you have the following information available:

- > The name of the Product that is being modified
- > The Batch Code associated with the Product that is being modified
- > A Product Name that identifies the Modification Product and is unique in the selected Batch Code (mandatory). The maximum length for a Product Name is 50 characters.
- > A description (free text) that provides additional information about the Modification Product, for example, the changes it includes (optional)
- > The details of the required changes, including Features to be added or removed, memory and license term updates, or any combination of these.

## Specifying License Terms and Memory for a Modification Product

To change the license terms for each Feature in the Modification Product, you can:

- > Change the expiration date for an **Expiration Date** license type by adding or subtracting days from the original expiration date. If the existing expiration date has already passed, the new expiration date will be the current date plus the number of days that you add to the Feature. (The expiration date for a **Time Period** license type cannot be modified.)
- > Change the number of remaining executions for an **Execution Count** license type by adding or subtracting a number of executions
- > Change the settings for concurrent instances, if appropriate.
- > Overwrite the license terms including selecting a new license type.
- > Change memory segments or data.
- > Cancel the license.

You can leave the license type value and the concurrent instances settings unchanged at this stage, and specify that they will be changed when each individual order for the Modification Product is processed.

## Example: Defining a Modification Product

**Scenario:** When the Product Manager of HQ Software originally defined the **HQ Design Pro** Product (in the example "[Example: Specifying License Terms and Protection Levels](#)" on page 105), the REPORT GENERATOR Feature was not yet available.

This Feature has now been developed, tested, and protected, and has been included in an enhanced version of **HQ Design Pro (v.2.0)**. This version of the Product is ready for sale to new customers, and can also be issued to customers who hold current licenses.

Accordingly, the Product Manager for HQ Software defines a Modification Product for the **HQ Design Pro** Product, named **HQ Design Pro v.2.0**.

When the Modification Product is defined, the REPORT GENERATOR Feature is added to the Product, with the same license terms as for the other Features.

## Issuing Modification Products

Modification Products can be included in orders in the same way as the original Products.

For example, if the Modification Product is intended to replace the Product in Sentinel protection keys that have already been deployed, it can be included in a *Protection Key Update* order. When the Protection Key Update is applied, the data for the Modification Product is added to the data for the original Product in the Sentinel protection keys.

For additional information on defining and producing orders, see ["Sentinel LDK Entitlements, Production, and Development Tasks" on page 122](#).

## Canceling Product Licenses

In certain circumstances, it may be necessary to cancel the license terms for one or more Features in a Product that has been delivered to a customer. For example:

- > To revoke a deployed license
- > To cancel the license for a Product that has been returned before its license terms have expired

A *Cancellation Product* can be defined for the Product, with values that cancel previous license terms. This Cancellation Product can be used whenever the license terms of the original Product need to be cancelled.

The process of canceling the license terms of a specific instance of a Product can include the following stages:

1. When the original Product needs to be cancelled, a Customer-to-Vendor (C2V file) is requested from the customer, containing the required license information.
2. An order for the Cancellation Product is defined and produced.
3. If the Product license is being moved to another computer, a new order for the original Product is produced with the appropriate details.
4. The changed license information is sent to the customer.
5. An acknowledgment receipt is returned by the customer when the change has been implemented.

For additional information on C2V files and on defining and producing orders, see ["Sentinel LDK Entitlements, Production, and Development Tasks" on page 122](#).

## Defining a Cancellation Product

Before you start to define a Cancellation Product, ensure that you have the following information available:

- > The name of the Product to be cancelled



- > The Batch Code associated with the Product to be cancelled
- > A Product Name that identifies the Cancellation Product and is unique in the selected Batch Code (mandatory). The maximum length for a Product Name is 50 characters.
- > A description (free text) that provides additional information about the Cancellation Product, for example, the reason it is required (optional)
- > The Features to be cancelled

## Specifying License Terms or Memory for a Cancellation Product

The options for defining the license terms for a Cancellation Product are exactly the same as for a Modification Product. For additional information, see ["Specifying License Terms and Memory for a Modification Product" on page 119](#).

### Example: Canceling a License

**Scenario:** A new customer, TOP Construction, purchased a one-year rental license for the **HQ Design Lite** Product. After three months, the customer wants to cancel the license and receive a refund.

HQ Software defines a Cancellation Product for the **HQ Design Lite** Product, with the license terms cancelled for all the Features in the Product. This Cancellation Product is only defined once—it can subsequently be used whenever required in similar circumstances.

TOP Construction is asked to send a Customer-to-Vendor (C2V) file. The file is received and processed in Sentinel LDK-EMS.

A Protection Key Update order is defined and produced for the **HQ Design Lite Cancellation** Product. The resulting Vendor-to-Customer (V2C) file containing the changed license details is sent to TOP Construction. TOP Construction applies the V2C file, then generates and returns a C2V file, confirming that the license cancellation has been applied. HQ Software then issues a refund.

For additional information on C2V and V2C files, and on defining and producing orders, see ["Sentinel LDK Entitlements, Production, and Development Tasks" on page 122](#).

# CHAPTER 11: Sentinel LDK Entitlements, Production, and Development Tasks

This section is intended for:

- > Users who are assigned the **Entitlement Manager** and **Production** roles in Sentinel LDK-EMS. It describes how to use Sentinel LDK-EMS to manage and produce entitlements (customer orders).
- > Users who are assigned the **Development** role in Sentinel LDK-EMS or who perform development-related tasks for Sentinel LDK and Sentinel LDK-EMS. For example: Creating bundles of Unlocked Products or exporting definition files.

For an overview of Sentinel LDK-EMS and for information on starting to use the application, see ["Introduction to Sentinel LDK-EMS" on page 90](#).

*In this section:*

- > ["Sentinel LDK Entitlement Processing and Production " below](#)
- > ["Managing Entitlements " on the next page](#)
- > ["Producing Entitlements" on page 131](#)
- > ["Performing Development-related Tasks" on page 136](#)
- > ["Enabling Trial Use and Grace Periods" on page 140](#)

**NOTE** This section provides high-level information on the entitlement management, production, and development-related processes in Sentinel LDK-EMS. For detailed practical instructions for using each function, see the [Sentinel LDK-EMS User Guide](#).

## Sentinel LDK Entitlement Processing and Production

An *entitlement* is the execution of a customer order for Sentinel LDK items, and can be one of the following:

- > An order for Products to be supplied with one or more Sentinel protection keys
- > A Protection Key Update that specifies changes to be made to the license terms and/or data stored in Sentinel protection keys that have already been deployed

For entitlements that generate Product Keys, the customer receives an email from Sentinel LDK-EMS that contains the keys. The customer can log in to the Sentinel EMS Customer Portal using the Product Key and activate the Products.

After Features and Products have been defined in Sentinel LDK-EMS, entitlements can be processed by:

- > ["Managing Entitlements " below](#)
- > ["Producing Entitlements" on page 131](#)
- > ["Performing Development-related Tasks" on page 136](#)

## Managing Entitlements

---

This section is intended for users assigned the **Entitlement Manager** role.

From the **Entitlements > Entitlements** tab in Sentinel LDK-EMS, entitlement managers can:

- > View the details of all entitlements associated with the selected Batch Code
- > Define and manage customers and channel partners
- > Define and manage entitlements
- > Process Customer-to-Vendor (C2V) information

For additional information on Batch Codes, see ["Personalized Vendor and Batch Codes" on page 21](#).

## Defining Entitlements

### Preparation for Creating an Entitlement

Before you start to define an entitlement in Sentinel LDK-EMS, ensure that you have the following information available:

- > (Optional) Details of the customer who placed the order
- > The Products to be included in the entitlement, including any details that must be finalized, such as the number of activations allowed, license model details, and memory file data
- > The production requirements, according to the type of entitlement:
  - Entitlement for Sentinel HL keys
  - Entitlement for Product Keys
  - Entitlement for Protection Key Update
- > Additional entitlement information (optional)

Sentinel LDK-EMS generates a unique entitlement ID (*EID*) for each entitlement that is created.

## Defining the Customer for the Entitlement

When you define an entitlement in Sentinel LDK-EMS, you can specify the customer who placed the order. You can search for an existing customer using the customer name or other identifying details, or you can define a new customer. (You can also define a new customer using the Customers page.)

## Including Products in the Entitlement

An entitlement can contain one or more Sentinel LDK Products. All Products are associated with a Batch Code. You select the Batch Code before you start to create a new entitlement.

**NOTE** Unlocked Products are not available for inclusion in entitlements. The process of generating files containing Unlocked Products is a Development task. For additional information, see ["Creating Bundles of Unlocked Products" on page 136](#).

Each Product is defined with a ["Locking Type" on page 377](#). The locking type determines the level of Sentinel LDK protection and the type of Sentinel protection key that can be supplied with the Product. The locking type assigned to a Product may determine the type of entitlement that can be produced:

- > Products defined only with the **HL** locking type can be included in entitlements for Sentinel HL keys, Product Keys, or for Protection Key Updates.
- > Products defined only with the **SL AdminMode** or **SL UserMode** locking type can be included only in entitlements for Product Keys or for Protection Key Updates.
- > Products defined with the **HL or SL AdminMode** or **HL or SL AdminMode or SL UserMode** locking type can be included in entitlements for Sentinel HL keys, Product Keys, or for Protection Key Updates

You cannot add a Product defined only with the **HL** locking type and another Product defined only with the **SL** locking type (whether **AdminMode** or **UserMode**) to the same entitlement.

For additional information on locking types, see ["Choosing the Protection Level for Your Products" on page 101](#).

## Specifying License Term Values

When a Product is initially defined in Sentinel LDK-EMS, the exact license term values for each Feature can be left unspecified. This enables you to include the same Product in different entitlements with different license term values.

In this case, the license values must be specified when each entitlement for the Product is processed.

You may be required to specify one or more of the following license term values for Features when processing an entitlement:

- > The date on which the license expires
- > The maximum number of times that the Feature can be used
- > The number of days until the license expires

You may also be required to specify the number of concurrent instances for one or more Features. This value specifies the number of instances of simultaneous usage that the license allows on the customer's network. Concurrent instances may relate to the network, processes, or machines.

An entitlement can be produced only after the license term values have been specified for all the Features in every Product included in the entitlement.

## Specifying Protection Key Memory Data

When a Product is initially defined in Sentinel LDK-EMS, memory data can be left unspecified. This enables you to customize memory data for each Product when defining the entitlement. For example, customer-specific memory data can be added to the Product when an entitlement is being processed.

## Specifying an Entitlement for Sentinel HL Keys

When an entitlement for Sentinel HL keys is produced, the ordered Products are programmed (burned) on one or more Sentinel HL keys to be shipped to the customer. For additional information on Sentinel HL keys, see ["Sentinel HL Keys" on page 26](#).

When you define the entitlement, you must specify the total number of Sentinel HL keys to be produced for the entitlement.

## Specifying an Entitlement for Product Keys

An entitlement for Product Keys enables you to produce activation strings for Sentinel protection keys.

The Products in the entitlement are associated with one or more Sentinel LDK Product Keys. A Product Key is a string of characters generated by Sentinel LDK-EMS and stored in a file for delivery to the customer.

After the end user receives the Product Key and returns it as proof of purchase, Sentinel LDK-EMS validates the Product Key and produces a Sentinel protection key. The Sentinel protection key is then sent back with the license terms and installed on the end user's computer, enabling the Product to be activated.

When you define an entitlement for Product Keys, you must specify the following information:

- > The number of Product Keys to be produced for the entitlement
- > The number of activations allowed for each Product Key. This is the number of machines on which each Product Key can be used.

While it is mandatory to use Product Keys for activation of software locked to Sentinel SL keys, Product Keys can also optionally be used for activating software that is locked to Sentinel HL keys.

**NOTE**

- > Before a Sentinel SL key can be used on an end user's computer, an unlocked trialware Product is typically installed on the computer. When the unlocked trialware Product is installed, it initializes the Sentinel LDK Run-time Environment, which is required for communication between the Sentinel SL key and the software.
- > The process of generating files containing unlocked trialware Products is a Development task. For additional information, see ["Creating Bundles of Unlocked Products" on page 136](#).

## Specifying a Protection Key Update Entitlement

A Protection Key Update entitlement specifies changes to be made to the license terms, Products, and/or data stored in Sentinel protection keys that have already been deployed to end users. A Protection Key Update can be applied remotely to Sentinel HL keys or Sentinel SL keys as follows:

- > By accessing the Sentinel LDK-EMS Customer Portal. Any outstanding updates are automatically applied for all protection keys installed on or connected to the customer's machine.
- > Using the Sentinel Licensing API by calling the **Update** function
- > By using the Sentinel Remote Update System utility
- > (For SL AdminMode keys) By placing the file that contains the update information in the appropriate directory on the end user's computer.

When the Protection Key Update entitlement is produced, a file containing the details of the changes is generated for each Sentinel protection key to be updated.

This file can be one of the following:

- > An executable file (EXE) that can be delivered to end users for use as instructed by your company
- > A Vendor-to-Customer (V2C) file that end users can process using the Sentinel Remote Update System utility (*RUS utility*)

For additional information on the RUS utility, see ["Sentinel Remote Update System \(RUS\)" on page 147](#). For additional information on updating SL AdminMode keys, see ["Applying License Updates to SL AdminMode Keys" on page 135](#).

When you define a Protection Key Update entitlement:

- > You must specify the total number of Sentinel protection keys to be updated as a result of this entitlement.
- > You may also need to select the specific Sentinel protection keys to be updated. For example, the entitlement may be for an organization with 100 Sentinel protection keys, and this entitlement is required to update the keys for only 10 specific users.

In Sentinel LDK-EMS, you can:

- Display a list of the customer's Sentinel protection keys

- View the contents of each key
- Select the keys to be updated

**NOTE** You cannot select more Sentinel protection keys than the total number of product keys specified in the **Product Details** area in the New Entitlement screen.

## Optional Entitlement Information

You can add the following optional information to the entitlement:

- > Order reference information that can identify the order in a different system, for example, an order number in your company's ERP system.
- > A comment that provides additional information about the order.

## Adding the Entitlement to the Production Queue

After you have specified all the necessary information for an entitlement, you can produce it immediately or "queue" it to add it to the *production queue*. The *queue* is a list of all entitlements that are awaiting production.

Entitlements in the production queue can be selected for production according to the criteria determined by your organization.

Sentinel LDK-EMS enables you to save as "draft" any entitlement that have not been completely defined, without losing the information that you may have already specified. You can open the entitlement and continue to define the entitlement details when convenient.

## Entitlement Status Values

During the course of its life cycle, an entitlement is assigned statuses as follows:

User Action	Resulting Entitlement Status	Description of Status
Create a new entitlement, click <b>Save</b> OR Re-open an entitlement.	<b>Draft</b>	Indicates that the entitlement is not yet ready for production. The entitlement details can be modified, or the entitlement can be deleted.
Create a new entitlement or edit an existing entitlement, click <b>Queue</b> .	<b>Queued</b>	Indicates that the entitlement is in the production queue, awaiting production. The details of a Queued entitlement cannot be changed. You can, however, remove the entitlement from the production queue by reopening it. This changes the status of the entitlement to Draft.

User Action	Resulting Entitlement Status	Description of Status
In an entitlement for Product Keys, select one or more Products and click <b>Produce</b> .	<b>Product Keys Generated</b>	Indicates that Product Keys for one or more Products in the entitlement have been generated. If the entitlement contains customer information, the customer receives an email. The email contains the Product Keys and information on how to log in the Sentinel LDK-EMS Customer Portal and activate the protection key.
	<b>Produced</b>	In an entitlement that includes multiple Product Keys, at least one Product Key has been used to activate the protected software. The entitlement contains additional Product Keys that have not yet been used.
	<b>Completed</b>	In an entitlement for protection key updates or for HL keys, the entire entitlement has been produced. In an entitlement for Product Keys, all the Product Keys have been used to activate the protected applications.
	<b>Acknowledged</b>	The end user has verified that the entitlement was applied at the customer site.

## Processing C2V Information

C2V files contain protected information about the license terms and data stored in deployed Sentinel protection keys. They do not contain private customer information.

C2V files can be generated using the Sentinel Remote Update System utility (*RUS utility*). For additional information on the RUS utility, see ["Sentinel Remote Update System \(RUS\)" on page 147](#).

C2V information stored in Sentinel HL keys and in C2V files can be retrieved for use in connection with Protection Key Update orders.

When a C2V file or Sentinel HL key is received from a customer, you must *check in* the information, in order to make the data in the file or key available to Sentinel LDK-EMS. The process of checking in the C2V information stores the data securely in Sentinel LDK-EMS, and enables you to view some of the information.

When you check in a C2V file, you can view the identifying information for the Sentinel protection key associated with the file, including the Batch Code, ID and key type. You can also view the Product details contained in the file. When you check in a Sentinel HL key, you can view similar information.



**NOTE** If you attempt to check in a C2V file for a Sentinel SL key, and Sentinel LDK-EMS detects that it has come from a cloned machine, you will not be able to check the C2V file into the database. For additional information about dealing with cloned Sentinel SL keys, see ["Specifying the License Terms for Features in a Product" on page 110](#).

## Formatting a Sentinel HL Key

You can format a Sentinel HL key to make it available for reuse. The process of formatting a Sentinel HL key deletes any orders that have been defined for the key but not yet produced. It also produces a V2C file that contains Protection Key Update information to be applied to the key using the RUS utility. Applying the Protection Key Update erases all license and memory data stored in the key.

## Order Processing and Production Examples

In the examples in this section, HQ Software defines the following orders for its customers:

1. Order for Sentinel HL keys
2. Order for Product Keys (Sentinel SL keys)
3. Protection Key Update order

### Order Example 1: Order for Sentinel HL Keys

**Scenario:** A new customer, ABC Design, orders the **ThalesCAD Office** Product from HQ Software with a license for 20 users.

Since the **ThalesCAD Office** Product is defined with Sentinel HL key protection, the details for this order are defined as follows:

- > **Customer:** ABC Design
- > **Product:** ThalesCAD Office
- > **Order type:** Sentinel HL keys
- > **Number of keys:** 20

When this order is produced, the **ThalesCAD Office** Product license is programmed on 20 Sentinel HL keys, which are then shipped to the customer.

### Order Example 2: Order for Product Keys (Sentinel SL Keys)

**Scenario:** On March 15, 2017, another customer, JL Optics, orders the **ThalesCAD Home** Product, with a license for use on two computers.

The **ThalesCAD Home** Product is defined with Sentinel SL key protection and an annual rental license. To ensure that the customer enjoys a full year's licensed use, the expiration date needs to be specified when the order is placed.

The details for this order are defined as follows:

- > **Customer:** JL Optics
- > **Product:** ThalesCAD Home
- > **Expiration date for DRAW and SAVE:** March 15, 2018
- > **Order type:** Product Key-based
- > **Number of Product Keys:** 1
- > **Number of Activations per Product Key:** 2

**NOTE** This example assumes that JL Optics has installed and used the **ThalesCAD Home [Trial]** unlocked trialware Product on the two computers before ordering the **ThalesCAD Home** Product. As a result, the Sentinel LDK Run-time Environment for Sentinel SL has already been initialized on those computers.

When this order is produced, a file is generated containing a Product Key. HQ Software sends this file to JL Optics by email.

Two end users at JL Optics open the file and enter the Product Key as required on the HQ Software Web site. The HQ Software customer interface application sends the Product Key to Sentinel LDK-EMS, which validates the Product Key and returns a Sentinel SL key to the customer.

The Sentinel SL key is installed on the two computers at JL Optics with the license information, and the **ThalesCAD Home** Product can be activated under the terms of the license.

## Order Example 3: Order for Protection Key Update

**Scenario:** HQ Software informs ABC Design that a new version of **ThalesCAD Office** has been released, containing the REPORT GENERATOR Feature, and that an upgrade is available for purchase. ABC Design orders the enhanced Product for five of its 20 users.

HQ Software has defined a Modification Product for the new version, **ThalesCAD Office v.2.0**. This Product is ready for inclusion in customer orders.

Before defining the Protection Key Update order, HQ Software needs to receive C2V files for the five Sentinel HL keys to be updated. ABC Design uses the RUS utility to generate the required C2V files and sends them to HQ Software.

After the C2V files have been received and checked in, HQ Software defines a Protection Key Update order for the Modification Product.

The details for this order are defined as follows:

- > **Customer:** ABC Design
- > **Product:** ThalesCAD Office v2.0
- > **Order type:** Protection Key Update

- > **Update delivery method:** `Manual`
- > **Number of Sentinel protection keys to be updated:** 5

During the order definition process, the five Sentinel HL keys to be updated are selected from all the keys issued to ABC Design, according to the C2V files received.

When this order is produced, a V2C file is generated for each selected Sentinel HL key and sent to the customer.

The selected five end users install the update on their Sentinel HL keys, using the RUS utility. They are then able to activate the upgraded version of **ThalesCAD Office** and to generate tailored reports.

## Producing Entitlements

This section is intended for Sentinel LDK-EMS users assigned the **Entitlement Manager** or **Production** role.

You can perform the following production tasks using the Entitlements page:

- > Produce Entitlements
- > View entitlements
- > Delete entitlements

**NOTE** If you have been assigned both the **Entitlement Manager** and the **Production** roles, you can produce an entitlement immediately after you finish defining it.

The process of producing an entitlement is determined by the type of entitlement:

- > Order for Sentinel HL keys
- > Order for Product Keys
- > Order for Protection Key Update

While producing any entitlement, you can open the entitlement and view its details.

## Producing Sentinel HL Key Entitlements

Before you start to produce an entitlement for Sentinel HL keys, Sentinel LDK-EMS enables you to prepare the appropriate Sentinel HL keys to use for the entitlement, by displaying:

- > The Sentinel HL key types that are valid for the entitlement
- > The number of Sentinel HL keys to be produced, as specified in the entitlement

Sentinel LDK-EMS determines which Sentinel HL keys are valid for the entitlement according to a number of factors, including:

- > The license terms defined for the Features in the Products included in the entitlement
- > The data defined in memory for each Product

- > The space required on the key to accommodate the entitlement

For example, if the license terms for a Product in the entitlement are based on one (or both) of these factors:

- > A number of days or an expiration date
- > A number of concurrent instances in a network environment

The entitlement can be produced only on Sentinel HL keys that support time-based licenses or support concurrency licenses. Most Sentinel HL (Driverless configuration) keys support these types of license terms.

For additional information about Sentinel HL key types and their capabilities, see ["End-User Keys" on page 26](#).

## Producing Entitlements for Product Keys

When you produce an entitlement for Product Keys, a TXT file is generated containing the Product Keys.

Before you generate the file, you must specify its required location, or accept the default location. The file is saved in the format **Product\_Keys\_[order ID].txt**.

After the file has been generated, the Product Keys are available for use. If customer information was provided in the entitlement, an email containing the Product Keys is generated automatically and sent to the customer.

## Producing Protection Key Update Entitlements

The entitlement production process generates a file containing the Protection Key Update information for each Sentinel protection key to be updated. After the file has been generated, it can be sent to the customer.

The V2CP (*Vendor-to-Customer*) file generated can be processed by the end user using the Sentinel Remote Update System utility (*RUS utility*)

For additional information on the RUS utility, see ["Sentinel Remote Update System \(RUS\)" on page 147](#).

**NOTE** An end user can apply all outstanding updates for the protection keys on their machine simply by accessing the Sentinel LDK-EMS Customer Portal and clicking the **Update Licenses** button. It is not necessary for the end user to receive the V2CP file. For more information, see ["Customer Portal - Applying Updates to Protection Keys" on the next page](#).

## Withdrawing and Deleting Entitlements

Under certain circumstances, you may need to withdraw an entitlement before it has been produced, or if it has been only partly produced. For example: If the customer cancels the order or significantly changes the order requirements.

If the entitlement is not yet in the production queue (**Queued** status), you can delete it. An entitlement cannot be deleted after it has been added to the production queue. You can, however, remove the entitlement from the production queue by reopening it. This changes the status of the entitlement to **Draft**.

A **Draft** entitlement is no longer available for production, but its content are available to view for reference.

You can delete an entitlement that is marked as a **Draft**. For additional information on editing and deleting entitlements, see the [Sentinel EMS User Guide for Sentinel LDK](#).

## Customer Portal - Activating Entitlements

If an entitlement includes customer information, then at the time Product Keys are generated, an email is automatically sent to the customer. The email contains the Product Keys and a link to the Sentinel LDK-EMS Customer Portal in Sentinel LDK-EMS.

To log in to the Customer Portal, customers click the provided link. At the login screen, the customer enters a Product Key.

If you specified in the entitlement that user registration is desired (or mandatory), the customer is requested (or required) to fill out a registration form.

Next, the Product Key screen is displayed.

The screenshot shows the 'Sentinel LDK-EMS Entitlement Management System' interface. At the top right, there are links for 'Welcome', 'Logout', and 'Help'. The main content area displays the 'Product Key : 8488cd06-b3e2-46de-97de-087bfc0989ac' with 'Online Activation' and 'Offline Activation' buttons. Below this, a table lists details: Product Key (8488cd06-b3e2-46de-97de-087bfc0989ac), Customer Name (Customers Unlimited), E-mail (contact@cust-un...), Channel Partner (-), E-mail (-), Activations (3), Remaining Activations (3), Previous Activations (0), and Enabled (true). At the bottom, there is a table for 'Products' and 'Lock Type' with one entry: 'SafeNet Design ...' and 'HL or SL-AdminMode'.

This screen displays the status of the Product Key, including the number of activations remaining.

The customer uses this screen to activate the entitlement as follows:

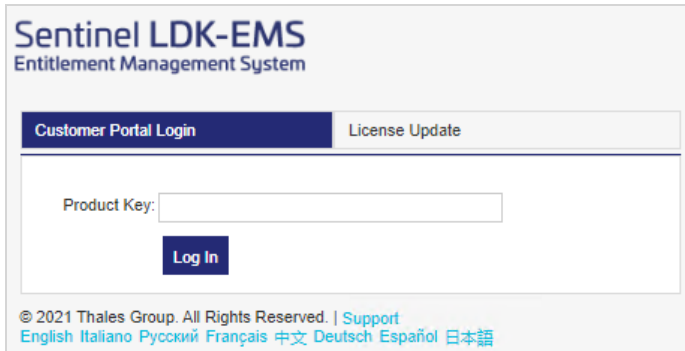
- > If the customer logged in to the Customer Portal from the machine where the license should be installed, the customer can click **Online Activation**. Activation of the entitlement proceeds automatically.
- > If the customer did not log in to the Customer Portal from the machine where the license should be installed, the customer can click **Offline Activation**. The customer can then download the RUS utility. The customer uses this utility in order to generate a C2V file and perform the activation process manually.

## Customer Portal - Applying Updates to Protection Keys

If an entitlement contains one or more updates to protection keys, several methods exist to apply the updates to the keys on the end user's machine.

One method is to send the generated V2C file to the end user by email. The end user can apply the V2C files using Admin Control Center or the RUS utility. You can also program a function in to the protected application to apply a V2C file using calls to Sentinel Licensing API or to Sentinel LDK-EMS Web Services.

A simpler method is to send the end user a link to the Customer Portal. The end user clicks the link from the machine where the relevant protection keys are located. The Customer Portal login screen is displayed.



On the **License Update** tab of the screen, the end user clicks **Update Licenses**. Sentinel LDK-EMS determines which protection keys are located on the end user's machine. For each protection key located, Sentinel LDK-EMS determine what update entitlements are outstanding for the key and applies all the entitlements in the correct sequence. No other action is required by the end user.

The following limitations exist for this method:

- > Sentinel LDK Run-time Environment version 7.100 or later must be present on the end user's machine.
- > This method is not applicable for an update that installs a new SL key on the end user's machine. The method can only be used to update existing keys.
- > This method cannot be used to update an SL Legacy key.

## Viewing License Updates

With certificate-based Sentinel SL keys, you can examine the sequence of updates that were applied to a Protection Key at a customer site. The V2C files that were applied to a protection key reside in a directory on the computer where the protection key is located. (A V2CP file is broken down into its component V2C files, and each V2C file is handled as described here.)

(For protection keys that were rehosted from a different computer, you can also examine the H2H files that contain rehost information.)

The licensing information in each V2C file is specified using XML tags. You can open any of these files using a simple text editor and read the contents. If you examine the files for a specific Protection Key ID in sequence, you can follow the history of updates that were applied to the key.

**NOTE** Information regarding SL legacy protection keys is not available in these files.

The V2C files on a given computer can be found in the following locations:

- > For SL AdminMode keys:

Windows	<b>%ProgramFiles(x86)%\Common Files\SafeNet Sentinel\ Sentinel LDK\installed\[vendorID]\</b> (For Windows x86, in: %ProgramFiles%\...)
Linux, Mac	<b>/var/hasplm/installed/[vendorID]/</b>

- > For SL UserMode keys:

Windows	<b>%ProgramData%\SafeNet Sentinel\ Sentinel LDK\installed\[vendorID]\</b>
Linux	<b>\$HOME/.hasplm/installed/[vendorID]</b>

The naming convention for the files is as follows:

<b>keyID_provisional.v2c</b>	Unlocked Product
<b>keyID_base.v2c</b>	Base Product
<b>keyID_updateX.v2c</b>	Update to a Base Product. Updates are numbered sequentially.
<b>keyID_rehost.h2h</b>	Rehost of a protection key

*keyID* is the Protection Key ID.

#### NOTE

- > Do not remove or modify these files. If any of these files are removed or modified, the protection key may become invalid.
- > The naming convention for V2C files can be modified in Sentinel LDK-EMS. For details, see the [Sentinel LDK-EMS Configuration Guide](#).

## Applying License Updates to SL AdminMode Keys

Several methods exist to apply updates to a Sentinel protection key (for example, using Sentinel Admin Control Center or Sentinel Licensing API). However, for SL AdminMode keys, an additional simplified method to apply updates exists.

**You can do either of the following:**

- > Use software to place the V2C file containing the license update directly into the **installed** directory described above (see ["Viewing License Updates" on the previous page](#)).
- > Instruct the end user to place the file into the **installed** directory.

The Sentinel License Manager detects the V2C file in the **installed** directory and automatically applies the license update.

**NOTE** These methods are not applicable for V2CP files.

If the license update is applied successfully, the Sentinel License Manager then moves the V2C file from the **installed** directory to the appropriate **installed\vendorID\** subdirectory.

If the license update installation fails, the Sentinel License Manager moves the V2C file to a separate directory called **invalid**. The failure is recorded in the Admin Control Center access log. The log can be viewed from the **Access Log** option in Admin Control Center.

## Performing Development-related Tasks

This section is intended for users of Sentinel LDK or Sentinel who perform development-related or administrative tasks.

This section describes the following development-related activities:

- > ["Creating Bundles of Unlocked Products" below](#)
- > ["Export Formats" on page 154](#)
- > ["Customizing and Branding the RUS utility" on page 138.](#)
- > ["Rate Limiting for Cloud Licensing \(Vendor-Hosted\)" on page 138](#)

### Creating Bundles of Unlocked Products

When a Product is defined in Sentinel LDK-EMS, it can be specified as an Unlocked Product that is not locked to a single machine. The Product can be further defined as an Unlocked Trialware Product for distribution as trialware or for use during a grace period.

Unlocked Products are often distributed as **bundles**. A bundle is an application that you create that installs one or more Unlocked Products. The bundle is typically executed as part of the installation procedure for the protected applications. The bundle enables a user to operate the protected applications for a restricted period of time (typically up to 90 days) or for a limited number of executions (typically 30).

**NOTE** Software that has been supplied with a trial license or for a grace period can be activated after a valid license is purchased, with either a Sentinel HL key or a Sentinel SL key.

**NOTE** The use of the Execution Count license type for Unlocked Products is only supported when working with Sentinel License Generation API.

For additional information on the purpose and use of unlocked trialware Products, see ["Designating Products for Trial or Grace Period Use" on page 104.](#)

Before you create a bundle of Unlocked Products, you should:



- > Generate a V2C file that contains the Unlocked Product licenses for the applications that you want to include in the bundle. This is done using Sentinel LDK-EMS or Sentinel License Generation API.
- > Generate a Run-time Environment installer that includes your vendor library. This is done using the Sentinel LDK Master Wizard. For details, see [Sentinel LDK Installation Guide](#).
- > Sign the Run-time Environment installer. For details, see ["Signing the Run-time Environment Installer" on page 196](#).

Create the bundle application. This application should perform the following operations on the user's machine:

1. Place the Run-time Environment installer in a temporary directory.
2. Place the V2C file in the same temporary directory as the Run-time Environment installer.
3. Execute the Run-time Environment installer. The installer automatically applies the V2C file after installing the Run-time Environment.
4. Remove the temporary directory and its contents.

After this process completes, the user can execute the protected applications within the limits of the trialware licenses or grace period.

**NOTE** When a bundle of Unlocked Products is installed on an end user's computer, a provisional Key ID is generated for the SL key. If a fully-licensed Product is installed on the computer where the bundled Unlocked Product exists, a new key ID is created in addition to the original provisional key ID.

## Exporting Definition Data

You can export data about Features, Products, vendors, and other information in various file formats. This information can then be used for development, protection backup, and other purposes. You can also export metadata for use in Admin Control Center.

To perform this action, you must be assigned the **Development** role .

In the Sentinel LDK-EMS **Developer** tab, you can use the **Export Definitions** function to produce the following output file types:

- > Metadata in Admin Control Center format
- > Features and Products in a C-style header file
- > Features and Products in a CPP-style header file
- > Features and Products in XML format
- > Features in CSV format

For examples of the output file contents, see the Sentinel LDK-EMS help system.

Before you export the Features, you must select the required Batch Code, specify the required file type, and define the name and location for the file.

As your software develops and additional Features are defined, you can use the **Export Definitions** function whenever you want to retrieve the data definitions from Sentinel LDK-EMS.

## Customizing and Branding the RUS utility

The RUS utility is a tool that can be distributed to end users to enable secure, remote updating of the license and memory data of Sentinel protection keys after they have been deployed.

End users can invoke the RUS utility in order to generate a C2V file.

Before you distribute the RUS utility, you must customize it with the Batch Code associated with the Sentinel protection keys that you have deployed to your end users, in order to enable them to generate C2V files, or to process files containing V2C information.

In addition, you can brand the text that is displayed to an end user when the RUS utility is opened. For example, you may want to display your company name and information about your software.

**Sentinel RUS Generator** is a tool that enables you to associate the RUS utility with your Batch Code. You can also use the simple HTML editor provided to enter, format, and preview the text to be displayed in the RUS utility. For more information, see [Sentinel RUS Generator](#).

For additional information on the RUS utility, see "[Sentinel Remote Update System \(RUS\)](#)" on page 147.

## Rate Limiting for Cloud Licensing (Vendor-Hosted)

Sentinel Licensing API supports rate limiting for cloud licensing. As a result, it is possible to implement rate limiting for cloud license API calls by protected applications on customers' machines. The use of rate limiting prevents overloading the license server and improves the licensed user experience if licensed user interactions with the applications are generating an excessive number of API calls to the license server.

Sentinel LDK enforces rate limiting using an identity-based policy using a token bucket algorithm ([https://en.wikipedia.org/wiki/Token\\_bucket](https://en.wikipedia.org/wiki/Token_bucket)), defining the use of buckets of tokens in the cloud license manager.

The cloud license manager uses the bucket as follows:

- > Each client identity that runs protected applications using cloud licensing is assigned a bucket. The bucket is assigned a starting number of tokens. The number assigned is also the maximum number of tokens that the bucket can contain.
- > New tokens are added periodically to the bucket.
- > Each of the following types of Licensing API calls consumes one or more tokens from the bucket and is affected by exceeding the rate limit: `hasp_login`, `hasp_logout`, `hasp_encrypt`, `hasp_decrypt`, `hasp_read`, `hasp_write`, `hasp_get_rtc`, and `hasp_update_session`. These calls can fail with the error `HASP_IDENTITY_RATE_EXCEEDED`.

Each Licensing REST API consumes one token from the bucket.

- > When the bucket is empty, the rate limit is considered to be exceeded.

The rate limit forces you to implement protection of your application using a limited number of API calls.

## Implementing Rate Limiting

To implement rate limiting for a cloud license manager hosted on your own server, configure the following parameters in the **hasplm.ini** file for the license manager:

<b>rate_token_max</b>	Maximum number of tokens in the bucket. This is also the starting value for the number of tokens in the bucket. If <b>rate_token_max</b> or <b>rate_token_period_ms</b> is 0, rate limiting is disabled.
<b>rate_token_period_ms</b>	Interval (in milliseconds) at which new tokens are added to the bucket. For example, a value of 3,000 means that a new token is added every 3 seconds.

For details, see [Sentinel LDK Installation Guide](#).

## Complying With Rate Limits

To ensure that API calls from your licensed users do not exceed the rate limits that you implement, you can configure Sentinel Licensing API or Sentinel Licensing REST API to limit the frequency with which API calls are sent to the cloud license manager. (Rate limiting only applies to API calls that use client identities.)

Best practices for complying with the rate limits include:

- > Avoid unnecessary but repetitive API calls, such as calling encrypt/decrypt every few second to keep a session alive.
- > Set the interval for periodic background checks for a valid license (using Sentinel Licensing API or Sentinel LDK Envelope) to a minimum of 10 minutes.
- > Configure the cloud license manager to allow automatic detaching of licenses. This offloads the burden of responses to API calls to the users' machines.

For more information, see [Sentinel Licensing API Reference](#).

## Rate Limiting Mechanism

The following mechanisms exist for implementation of rate limiting:

- > With Sentinel Licensing API 9.12 or later:

The bucket is stored in the identity session. This means that each application uses a different bucket, and different applications do not interfere with each other. The license server always fulfills the API calls, but if the rate limit is exceeded, the license server notifies the Licensing API how long to wait before making a new call. The Licensing API then causes the API call to fail on the client side, without any connection with the license server, until the stated time has elapsed.

- > With Sentinel Licensing API 8.51 or earlier:

It is not possible to make the API call to fail on the client side. Therefore, the license server makes the API call fail and returns the error `HASP_IDENTITY_RATE_EXCEEDED` to the client. The Licensing API also returns the error `HASP_IDENTITY_RATE_EXCEEDED`.

**NOTE** This method of failing the API call is inefficient. It saves only a fraction of the server work, as the server still has to process the API call. The older Licensing API also consumes additional API calls.

> With Sentinel Licensing REST API:

Once rate limiting is triggered, the license server makes the API call fail and returns the error `HASP_IDENTITY_RATE_EXCEEDED` at the LDK level, and returns the error **429** at the HTTP level. A `Retry-After` header is included to this response, indicating how long to wait before making a new API call.

## Enabling Trial Use and Grace Periods

This section provides examples that demonstrate the use of unlocked trialware Products:

- > To distribute a Product for use on a trial basis for a limited period
- > To enable use of a licensed Product during a grace period

### Example 1: Issuing an Unlocked Trialware Product for Trial Use

**Scenario:** HQ Software decides to offer visitors to their Web site the option of downloading and using their **HQ Design Demo** Product for 30 days.

When the original licensing plan definitions were implemented, the **HQ Design Demo** Product was defined as an Unlocked Product.

The license terms for the Features in the Product were set to **Time Period** with a value of **30** days.

A V2C file containing the **HQ Design Demo** Unlocked Product is generated. HQ Software creates a bundle to install the Unlocked Product and the Run-time Environment.

The HQ Software Web master adds the bundle to the Web site, with download and execution instructions for potential trial users.

### Example 2: Issuing a Product for a Grace Period

**Scenario:** A new customer, XYZ Construction, has purchased a 50-user license for the **HQ Design Pro** Product, which is available only with Sentinel HL key protection. The Sentinel HL keys are being prepared and shipped, but meanwhile the customer wants to start using the **HQ Design Pro** Product immediately.

HQ Software needs to enable XYZ Construction to activate and use the **HQ Design Pro** Product during a grace period, until the Sentinel HL keys arrive and are distributed to the end users.

For this purpose, a version of the **HQ Design Pro** Product is defined as an Unlocked Product, with the Product name **HQ Design Pro Grace**. The PRINT REPORTS Feature is removed from this version. The license terms for the remaining four Features are set to **Time Period** with a value of **30** days.

A V2C file containing the **HQ Design Pro Grace** Unlocked Product is generated. HQ Software creates a bundle to install the Unlocked Product.

The bundle file is sent to the customer, for distribution to the end users. End users can run the bundle, which installs the Sentinel LDK Run-time Environment and the **HQ Design Pro Grace** Unlocked Product on their computers. They can then use the program for 30 days until they receive their Sentinel HL keys and can activate the full Product.

# CHAPTER 12: Sentinel LDK Administration and Customer Services

This section discusses the following topics:

- > **Administration tasks.** This topic is intended for users authorized to perform Sentinel LDK Administration tasks. The topic describes how to use Sentinel LDK-EMS to define user details, manage Sentinel LDK licenses and the Sentinel LDK Master license, and configure system settings.
- > **Customer Services.** This topic is intended for users authorized to perform Sentinel LDK-EMS Customer Services tasks. It describes how to use Sentinel LDK-EMS to view and edit customer details, and to perform manual Product activation for customers.
- > **Channel partners.** This topic provides an overview of the functionality that is available in Sentinel LDK-EMS for working with channel partners.

For an overview of Sentinel LDK-EMS and for information on starting to use the application, see ["Introduction to Sentinel LDK-EMS" on page 90](#).

*In this section:*

- > ["Administration Tasks" below](#)
- > ["Customer Services" on page 145](#)
- > ["Channel Partners" on page 145](#)

**NOTE** This section provides high-level information on the Administration, Customer Services, and channel partner processes in Sentinel LDK-EMS. For detailed practical instructions for using each function in Sentinel LDK-EMS, see the [Sentinel LDK-EMS User Guide](#).

## Administration Tasks

After you first install Sentinel LDK in your organization, you can log in to Sentinel LDK-EMS using the default user name and password (`admin`) provided for your use by Thales. By default, an admin user is authorized to perform all tasks in Sentinel LDK-EMS, including Administration tasks.

**NOTE** The 'admin' administrator details cannot be viewed or modified. Only the password can be changed.

After logging in to Sentinel LDK-EMS the first time, select the **Change Password** function at the top of the screen and change your user password as soon as possible.

To be able to use Sentinel LDK with your vendor-specific Batch Codes and license, you must first introduce one of the Vendor keys provided in your Starter Kit.

For additional information on Vendor keys, see ["Sentinel LDK Vendor Keys" on page 22](#).

For additional information on managing a Master key, see ["Maintaining Your Sentinel LDK Master Key" on the next page](#).

For additional information about the various modules in your Sentinel LDK Master license, see ["Understanding the Sentinel LDK Master License" on page 305](#).

From time to time, you will need to renew your Sentinel LDK license, or to replenish your pools of SL keys or network seat licenses. You can schedule email notifications to be sent when it is time to renew or reorder, ensuring you uninterrupted use of Sentinel LDK.

For additional information about configuring and scheduling email notifications, refer to the [Sentinel LDK-EMS User Guide](#).

**NOTE** If you are evaluating Sentinel LDK-EMS installed on-premises, you can use the provided **DEMOMA** Batch Code, which does not require a Master key.

You can now define additional Sentinel LDK users in your organization, including assigning the users the appropriate roles and authorizing access to Batch Code. For additional information, see ["Maintaining User Details" below](#).

## Maintaining User Details

In the Sentinel LDK-EMS **Users** tab (**Administration > Users**), you can view the details of all currently defined users and perform the following tasks:

- > Define Sentinel LDK users
- > Define Channel Partner users
- > Change user details and passwords
- > Control user access

## Defining Sentinel LDK Users

Before you start to define Sentinel LDK users, ensure that you have the following information for each new user:

- > The user name to be assigned to the user for the purpose of logging in to Sentinel LDK.
- > The password to be assigned to the user.
- > The user's email address.
- > The Batch Codes that the user is authorized to access.

- > The roles to assign to the user. For additional information on the functions authorized for each role, see ["User Types and User Roles in Sentinel LDK-EMS" on page 92](#).

## Changing User Details and Passwords

After you have defined a user, you can change any of the user's details.

Users can change their own passwords. If needed, you can change the password for a user without knowing the current password. This is useful if a user loses or forgets their password.

## Controlling User Access

In certain circumstances, you may want to prevent a user from logging in to Sentinel LDK. If the user has left the company, for example, or will no longer be using Sentinel LDK, you can delete the user details.

You can prevent or allow a user to access Sentinel LDK by clearing or selecting the **Login Allowed** check box.

## Maintaining Your Sentinel LDK Master Key

This topic is relevant only for Sentinel LDK-EMS installed on-premises.

When you select the **Administration > Master** tab in Sentinel LDK-EMS, you can view the details (for the selected Batch Code) of the Master key connected to the Sentinel LDK-EMS Server.

You can perform the following tasks using the Master page in Sentinel LDK-EMS:

- > Generate a C2V file for a selected Master key. You send this file to your Thales representative when you want to:
  - update your Master key modules
  - replenish your pool of SL keys
  - replenish your pool of HL or SL network seat licenses
- > Apply the V2C file returned to you by your Thales representative to the Master key. This updates the Master key with the contents of your order.
- > Specify Mail Notification properties

For more information on the Master key, see ["Understanding the Sentinel LDK Master License" on page 305](#).

## Introducing Sentinel Vendor Keys

Before you can work effectively, you must introduce one of your Vendor keys to Sentinel LDK-EMS using the Sentinel LDK Master Wizard. Introducing a Vendor key performs a number of important functions. For details, see ["Master Wizard" on page 13](#).

When working with Sentinel LDK-EMS on-premises: To generate licenses for your protected applications, your Master key must be connected to the machine on which Sentinel LDK-EMS is installed.



## Generating a C2V File

When you submit an order for an update to your Sentinel LDK Master Key licenses, regardless of whether it is to renew a license or to replenish your pools of SL keys or Network Seat licenses, you need to generate a C2V file for the Master key that is to be updated. You then send the C2V to your Sentinel LDK supplier, together with your order. The C2V file contains encrypted information about the current status of your Master key, including its unique ID.

## Defining Mail Notification Properties

You can specify who is to receive notifications that your Sentinel LDK Master Key licenses and pools of SL keys or Network Seat licenses are about to expire. In addition, you can define the thresholds after which the notifications are sent.

## Customer Services

---

If you have been assigned the **Customer Services** role, you can manage the list of customers — you can define customers, change customer details, and mark customers as obsolete.

You can enable or disable a Product key for a customer, or increase the number of activations available for a Product key.

If a customer is unable for any reason to activate a Product remotely, you can activate the Product manually for the customer, using the Product Key and a Customer-to-Vendor (C2V) file for the customer's Sentinel protection key. The output of the manual activation process is a Vendor-to-Customer (V2C) file that can be sent to the customer. You can request that the customer return a C2V file to confirm that the Product has been activated.

For additional information on C2V files, see ["Processing C2V Information" on page 128](#).

## Channel Partners

---

A channel partner is a company that partners with you to market and sell your products. Sentinel LDK-EMS enables you to allow your channel partners to access Sentinel LDK-EMS functionality to assist them in servicing their customers.

- > You can define channel partners on the **Customers > Channel Partners** page.
- > You can associate each channel partner with one or more Products from the catalog.
- > You can associate entitlements with a channel partner. When you open a screen to add Products to an entitlement, only Products associated with the channel partner are listed.
- > You can list all entitlements associated with a given channel partner.
- > You can designate users of Sentinel LDK-EMS as *Channel Partner users*. A Channel Partner user must be associated with a specific channel partner.

Each Channel Partner user is authorized to perform the following actions in Sentinel LDK-EMS for their associated channel partner:

- Create and manage their own end-user customers. Channel Partner users can see only their own customers. They cannot see the customers of other channel partners or customers of the software vendor.
- View, produce, and activate entitlements for their customers. Channel Partner users can see only committed entitlements that are associated with their channel partner.
- Resend emails for entitlements for their customers.
- Display product keys for entitlements for their customers.
- Check in, browse, and view details of C2V files for their customers.

To perform many of the activities associated with channel partner functionality, you must obtain the **Channel Partner** module for your Sentinel LDK Master license.

The table that follows indicates which activities require the Channel Partner module.

Activity	Requires Channel Partner Module
Define channel partner on the <b>Customers &gt; Channel Partners</b> tabbed page.	No
Associate an entitlement with a channel partner; list entitlements associated with a channel partner.	No
Associate a channel partner with one or more Products from the catalog.	Yes
Designate a user of Sentinel LDK-EMS as a Channel Partner user.	Yes
Log in to Sentinel LDK-EMS as a Channel Partner user and perform functions related to the relevant channel partner (described in this section). When a Channel Partner user opens a screen to add Products to an entitlement, only Products associated with the channel partner will be listed.	Yes

# CHAPTER 13: Sentinel Remote Update System (RUS)

This section describes the Sentinel Remote Update System utility (*RUS utility*) and explains how to use this utility to update license data remotely for deployed Sentinel protection keys.

**NOTE** Updates can be also applied to a deployed Sentinel protection key in any of the following manners:

- > An end user can apply all outstanding updates for the protection keys on their machine simply by accessing the Sentinel LDK-EMS Customer Portal and clicking the **Update Licenses** button. In this case, it is not necessary for the end user to receive the V2C file. For more information, see ["Customer Portal - Applying Updates to Protection Keys" on page 133](#).
- > The end user can use the Sentinel Admin Control Center (if present on the end user's machine) to apply a V2C file.
- > You can use the Sentinel Licensing API, by calling the **Update** function. For additional information, see the [Sentinel Licensing API Reference](#). For SL AdminMode keys, also see ["Applying License Updates to SL AdminMode Keys" on page 135](#).

*In this section:*

- > ["RUS Utility Overview" below](#)
- > ["RUS Workflow " on the next page](#)
- > ["Using the RUS utility" on page 149](#)

## RUS Utility Overview

The RUS utility is an advanced utility that enables secure, remote updating of the license and memory data of Sentinel protection keys after they have been deployed. As part of the basic concept underlying Sentinel LDK, the RUS utility facilitates ongoing licensing well after protection has been implemented. For additional information on Sentinel LDK concepts, see ["Protect Once—Deliver Many—Evolve Often" on page 20](#).

The RUS utility is used for the following:

- > The RUS utility provides a simple and secure method of updating your licenses remotely, after you have delivered your protected software together with the Sentinel protection keys. You simply need to update the

license and deliver update files to your customers.

- > The RUS utility enables you to receive information on the current status of Sentinel LDK licenses at your customers' sites, and to securely extend or reduce the functionality of these licenses, without recalling the Sentinel protection keys.
- > The RUS utility can be used by an end user to generate the fingerprint of a computer at your customer's site. You can use this fingerprint to generate a V2C (or V2CP) file for the customer. The customer can then use the RUS utility to apply the V2C (or V2CP) file and generate an SL key for the protected application.
- > The RUS utility can be used to transfer (*rehost*) an SL key from one of your customer's computers to another, without any intervention on your part. (An SL key can only be rehosted if this function was enabled by the vendor when the SL key was generated.)
- > All Sentinel protection keys except the Sentinel HL Basic key can be updated using RUS utility. However, for a disabled HL Basic key, an end user can use RUS utility to:
  - generate a C2V file containing information about the disabled key.
  - apply a V2C (or V2CP) file to re-enable the disabled key.

**NOTE** You must generate a customized version of the RUS utility for distribution to your customers. Customizing the RUS utility associates it with your [Batch Code](#). For additional information, see ["Customizing and Branding the RUS utility" on page 138](#).

Thales recommends that you bundle the customized RUS utility executable (`rus.exe`) with your protected application for distribution to end users.

## RUS Workflow

When you deliver your Products to a customer, you can include a customized version of the RUS utility with the installation package. You can also include the instructions for using RUS.

(To perform rehost, your customer will require the customized RUS utility.)

When a license update is required, you have the option of either retrieving customer licensing information from Sentinel LDK-EMS, or of requesting that a customer produces and sends you a Customer-to-Vendor (C2V) file for the Sentinel protection keys to be updated. C2V files have a `.c2v` extension and contain information on the licensing and memory content of the Sentinel protection keys.

When you receive C2V files from a customer, you check them in using Sentinel LDK-EMS. For additional information, see ["Processing C2V Information" on page 128](#).

Regardless of whether you obtain the data from Sentinel LDK-EMS or in the form of a C2V file from your customer, the collected data enables you to produce an update most suited to the customer's needs. At no point in this workflow is it necessary to reconfigure security or protection at the customer's site.

You define the requested license updates in Sentinel LDK-EMS as Protection Key Update orders for delivery to the customer. For more information on defining Protection Key Update orders, see ["Defining Entitlements" on page 123](#).

The process of producing a Protection Key Update order generates a file for each Sentinel protection key to be updated. This can be either a Vendor-to-Customer (V2C) file or an executable that contains the license update data. For more information on the Protection Key Update order production process, see ["Producing Protection Key Update Entitlements" on page 132](#).

The output file is then delivered to the end user, who either runs the executable as instructed by you, or uses the RUS utility to apply the license update data contained in the V2C file.

## Example: Using RUS for License Updates

**Scenario:** One of HQ Software's customers, ABC Design, has ordered the upgraded version of **HQ Design Pro** that contains the new REPORT GENERATOR Feature, for five of its 20 **HQ Design Pro** users. The customer is asked to send C2V files containing details of the five deployed Sentinel HL keys to be updated.

HQ Software included the RUS utility in the installation package for **HQ Design Pro**. ABC Design uses the RUS utility to generate the C2V files and sends them to HQ Software. These files contain the current status of the license on the specific Sentinel HL keys.

HQ Software checks in the C2V files, defines a Protection Key Update order for the **HQ Design Pro v.2.0** Modification Product, and produces a license update contained in five V2C files. For additional information on this example order, see ["Order Example 3: Order for Protection Key Update" on page 130](#).

The V2C files are sent by email to ABC Design. Each of the five end users applies the update to their Sentinel HL key using the RUS utility, and returns a C2V file containing a confirmation receipt.

## Using the RUS utility

The RUS utility window contains the following tabs:

- > **Collect Status Information:** The parameters in this tab are used to collect information on the current status of the licenses in the Sentinel protection key or collecting fingerprint information. The end user specifies a name and location for the generated C2V file. If more than one Sentinel protection key is installed, the user selects the required key. No private customer data is included in the C2V file.
- > **Apply License Update:** The parameters in this tab are used to apply a V2C file and update licenses in a Sentinel protection key.
- > **Transfer License:** The parameters in this tab are used on the source computer and recipient computer to rehost an SL key from the source computer to the recipient computer.

The topics that follow provide instructions that you can customize and send to your customers to perform various functions using the RUS utility.

- > ["Collecting Sentinel Protection Key License Data" on the next page](#)

- > ["Collecting Computer Data" below](#)
- > ["Applying an Update to a License" on the next page](#)
- > ["Rehosting an SL key" on the next page](#)

**NOTE** If you are using the RUS utility with a Sentinel HL key, (hardware-based key) you must connect the key before performing any of the following procedures. The RUS utility automatically locates any Sentinel SL keys (software-based keys) installed on your computer.

## Collecting Sentinel Protection Key License Data

You can use the RUS utility to produce a Customer-to-Vendor (C2V) file containing information on the current status of the licenses in your Sentinel protection keys. You can then send this file to the software vendor in order to receive a license update.

### To retrieve the current license information from a Sentinel protection key

1. Launch the RUS utility (`rus.exe`).
2. Click the **Collect Status Information** tab.
3. Ensure that **Update of Existing Protection Key** is select at the bottom of the screen.

**NOTE** If you are collecting information for a disabled HL Basic key, ensure that no other HL keys are connected to the computer.

4. Click **Collect Information**. The Save key status as window is displayed.
5. Specify the directory where you want to store the C2V file. Enter a file name and click **Save**.
6. If more than one Sentinel protection key is located, a list of the keys is displayed. Select the required key, or disconnect the keys that are not required, and click **Refresh**.
7. The C2V file for the Sentinel protection key is generated and saved in the required location. The file can now be sent for processing to produce an update.

## Collecting Computer Data

You can use the RUS utility to produce a Customer-to-Vendor (C2V) file containing information on the computer where you want to install a Sentinel protection key for a protected application. You can then send this file in order to receive a license update. This procedure would be used if a Sentinel protection key does not currently exist on the computer.

### To retrieve the current computer information:

1. Launch the RUS utility (`rus.exe`).
2. Click the **Collect Status Information** tab.

3. Ensure that **Installation of New Protection Key** is select at the bottom of the screen.
4. Click **Collect Information**. The Save key status as window is displayed.
5. Specify the directory where you want to store the C2V file. Enter a file name and click **Save**.
6. The C2V file for the Sentinel protection key is generated and saved in the required location. The file can now be sent for processing to produce a Sentinel protection key.

## Applying an Update to a License

You can use the RUS utility to apply an update to the licenses stored in your Sentinel protection keys.

### To update the licenses in Sentinel protection keys:

1. Launch the RUS utility (`rus.exe`) or double-click the Vendor-to-Customer (V2C or V2CP) file that you have received containing the update data.

**NOTE** If you have received an update as an executable, double-click the file and it will automatically launch RUS utility.

2. Click the **Apply License File** tab. (This might be the only tab displayed.)

## Rehosting an SL key

You can use the RUS utility to transfer an SL key from one computer (the *source computer*) to another (the *recipient computer*). This is a three-step procedure that uses the RUS utility on both computers. The two computers can be at different sites.

**NOTE** The following requirements apply:

- > The RUS utility must be present on both computers
- > The operating system on the recipient computer must be one that supports the Run-time Environment. For SL AdminMode keys and SL Legacy keys, the Run-time Environment must be present on the recipient computer.
- > If the recipient computer is a virtual machine:
  - The license must allow the protected application to operate on virtual machines.
  - The virtual machine must be one that is supported by Sentinel LDK.

### Step 1: Collect Information About the Recipient Computer

1. On the recipient computer, launch the RUS utility (`rus.exe`).
2. Click the **Transfer License** tab.

3. Follow the instructions labeled "Step 1" to collect information about the computer and save it to a file. Make sure that the file (or a copy of the file) is accessible on the source computer.

### Step 2: Generate the License Transfer File

1. On the source computer, launch the RUS utility (`rus.exe`).
2. Click the **Transfer License** tab.
3. Follow the instructions labeled "Step 2" to select the SL key to transfer, read the recipient information file, and generate a license transfer (h2h) file. Make sure that the license transfer file (or a copy of the file) is accessible on the recipient computer.

**NOTE** After you perform this step, the SL key is no longer available on the source computer. Be sure to keep a copy of the transfer file until you have completed the transfer procedure.

### Step 3: Apply the License Transfer File

1. On the recipient computer, in the RUS utility, click the **Apply License File** tab
2. In the **Update File** field, click the browse button and locate the license transfer (h2h) file.
3. Click **Apply Update**. The SL key is installed on the recipient computer.

**NOTE** To ensure the success of the transfer procedure, all the steps in the procedure should be completed within no more than a few days of the time you first start the process.



# CHAPTER 14: Generating Sentinel LDK Reports

This section describes the Reporting facility in Sentinel LDK-EMS.

*In this section:*

- > ["Managing Reports" below](#)
- > ["Permissions for Working With Reports" on the next page](#)
- > ["Scheduling Reports" on the next page](#)
- > ["Presentation Formats" on the next page](#)
- > ["Export Formats" on the next page](#)
- > ["Available Reports" on the next page](#)
- > ["Custom Reports" on page 155](#)

## Managing Reports

You can produce reports with valuable business information based on data in the Sentinel LDK-EMS database. This enables managers to obtain data for analyzing how their software is used and the purchasing preferences of their customers. The information can also be leveraged to maximize revenues from license renewals, to up-sell existing customers, and to turn trial users into buyers.

Sentinel LDK-EMS Reports connect directly to the Sentinel LDK-EMS database, and generate reports based on SQL queries. Both predefined and custom (user-defined) reports are available.

Sentinel LDK-EMS Reports can present information both in tabular and (where appropriate) graphical formats, and can export report data in a variety of formats for further processing and analysis.

You can perform the following tasks using the **Reports > Reports** tab in Sentinel LDK-EMS:

- > View the list of available reports
- > Define and generate reports

You can perform the following task using the **Reports > Scheduled Reports** tab in Sentinel LDK-EMS:

- > Manage the scheduling and distribution of Sentinel LDK-EMS reports that are to be run automatically on a regular basis

For detailed information on managing reports, see the [Sentinel LDK-EMS User Guide](#).

## Permissions for Working With Reports

---

Access to Report generation is limited to Sentinel LDK-EMS vendor users who have been granted the **Batch Code Admin** role or **Report Generation** role with the relevant permissions to view reports directly in Sentinel LDK-EMS. (The **Report Generation** role provides access only to the Reports area in Sentinel LDK-EMS for the specified Batch Codes.)

An authorized user can use the scheduling option to define a distribution list for each report. Each member of the distribution list receives the report by email. The list can include Sentinel LDK-EMS users (for whom an email address has been specified in Sentinel LDK-EMS) or any valid email address.

No special authorization is required to receive reports by email.

## Scheduling Reports

---

An authorized Sentinel LDK-EMS vendor user can generate and view reports on demand. In addition, the user can define a schedule for generation of each report and a distribution list of people to receive the report automatically by email each time the report is generated. Both predefined and custom reports can be scheduled.

Reports can be scheduled for generation and distribution based on a daily, weekly, or monthly scheduling definitions. A scheduled report can also be generated and distributed on-demand.

## Presentation Formats

---

All reports are generated in tabular (text-based) format. In addition, where relevant, each report includes a graphical presentation of the data, in either pie chart or bar chart format.

## Export Formats

---

Each report can be exported from Sentinel LDK-EMS or sent to the recipients in the distribution list in any of the following formats:

- > PDF
- > RTF
- > XLS
- > HTML
- > CSV (Comma-separated values)

## Available Reports

---

The following Sentinel LDK-EMS reports are available out-of-the-box.

Report Name	Report Description
<b>Entitlements</b>	
Customer Entitlement	Lists all entitlements for customers and channel partners.
Customer Activation	Lists activations by customer
Total Entitlement Utilization	Summarizes total and activated entitlements by Product
<b>Licenses</b>	
License Expiration	Lists all licenses due to expire within a specific period.
Most Popular Products Ordered	Indicates orders for Products during a given period by channel partner

## Custom Reports

Sentinel LDK-EMS enables you to define custom reports. This lets you design reports that satisfy the specific business requirements for your organization.

Custom Reports are defined by creating an SQL query that extracts the specific information you require from the Sentinel LDK-EMS database. For more information, select the **Administration > Custom Reports** tab in Sentinel LDK-EMS.

The Custom Reports utility is licensed separately from Sentinel LDK-EMS. To obtain a license to use Custom Reports, contact your Thales representative.

# CHAPTER 15: Cloud Licensing Using Sentinel LDK Cloud Portal

This section describes how to implement cloud licensing in Sentinel LDK when using Sentinel LDK Cloud Portal to manage licensed users.

**NOTE** To implement cloud licensing using Sentinel Admin Control Center, see ["Cloud Licensing Using Sentinel Admin Control Center" on page 171](#).

*In this section:*

- > ["Cloud Licensing Overview" below](#)
- > ["Implementation Summary" on page 158](#)
- > ["Setting Up the License Server Machine" on page 159](#)
- > ["Installing a Client Identity on an End User's Machine" on page 160](#)
- > ["Cloud Licensing Performance" on page 161](#)
- > ["Overview for Multi-Level License Servers" on page 162](#)
- > ["Online Detach: Setting Up Multi-Level License Server Machines " on page 165](#)
- > ["Offline Detach: Setting Up Multi-Level License Server Machines" on page 167](#)
- > ["Configuring High Availability for Cloud Licensing" on page 170](#)

## Cloud Licensing Overview

Cloud licensing enables end users to access local software with a license hosted in the cloud. You can set up a license server machine and grant access to cloud licenses over the Internet for your customers.

Cloud licensing uses identity-based access to give you granular control over who can access a network seat from an SL license.

Cloud licensing provides an alternative model for distributing licenses for protected applications to customers. Rather than providing SL protection keys to the customer, you host the SL licenses on your own license server and provides *client identities* for each end user. Each client identity is represented by a unique *identity string* that the end user installs on their machine.

The client identity entitles the user to execute an instance of the protected application, detach a network seat that can be used offline, or both. The user can install the identity string on multiple machines (if permitted by the vendor), but can only execute a single instance or detach a single network seat at any point in time.

Among the benefits of using client identities are the following:

- > Software can easily be used on different machines using a single identity.
- > Client identities are not affected by operating system upgrades or hardware upgrades.
- > You can easily disable a client identity if necessary.

Client identities can be assigned an expiration date. This provides the following additional benefits:

- > You can easily manage trial by providing users with expiring identities that consume the same license. This eliminates the need to generate a new key and a new license for each trial.
- > You can easily provide customers with an emergency cloud license in case they face issues with their local HL or SL key.
- > The IT administrator can manage which users can access a concurrency license and for how long.

Client identities are managed using Sentinel LDK Cloud Portal. described below.

For more information on the benefits of cloud licensing, see ["Understanding Cloud Licensing" on page 33](#).

**NOTE** Cloud licensing is only available if you have the [Cloud Licensing module](#) in your Sentinel LDK Master license at the time that you generate the SL licenses.

## Sentinel LDK Cloud Licensing Portal

---

You can manage cloud licensing and client identities using *Sentinel LDK Cloud Portal*.

Sentinel LDK Cloud Portal is a web-based application for software vendors who want to host cloud licenses (CL keys) on their own servers. Sentinel LDK Cloud Portal simplifies setting up and distributing client identities to licensed users located anywhere, granting them access to the protected applications running on their own machines.

Each client identity can be granted a license to all products in specific CL keys, or the license can be limited to specific products in the CL key.

Sentinel LDK Cloud Portal enables you to optionally offload the responsibility of managing client identities to designated users in each customer's organization. The designated users can log in to Sentinel LDK Cloud Portal to easily generate and deliver client identities, set and modify licensed user permissions, and revoke or restore access. Alternatively, you can continue to manage client identities yourself on behalf of customer organizations or for direct individual customers.

You also have the option of allowing customers to set up a second level of license servers to provide greater control over the distribution of network seats within an organization and to minimize the overhead of license administration. For more information, see ["Overview for Multi-Level License Servers" on page 162](#).

## Implementation Summary

The following steps summarize the process of implementing the cloud licensing model.

1. Set up a global license server that allows access to licenses via the Internet. For details, see ["Setting Up the License Server Machine" on the next page](#).
2. Deploy Sentinel LDK Cloud Portal in the same environment. For more information, see [Sentinel LDK Installation Guide](#).
3. In [Sentinel LDK-EMS Configuration Guide](#), see the description of the **URL for Cloud Portal** parameter and ensure that you comply with all the requirements.
4. Log in to Sentinel LDK Cloud Portal using the credentials set during deployment and modify the configuration parameters as required. For more information, see [Sentinel LDK Cloud Portal Guide for Vendors](#).
5. Protect your applications using Sentinel LDK. Ensure that the **Locking Type** includes the option to use SL AdminMode keys.
6. Use Sentinel LDK-EMS to generate a separate CL key for each customer organization, using the **Produce & Push** option to deliver entitlements.

The license must allow concurrency. Specify the required number of network seats.

7. For each customer organization defined in Sentinel LDK-EMS, use Sentinel LDK-EMS to define one or more contacts. Each contact can serve as a customer administrator in Sentinel Cloud Portal to define and manage licensed users within their organization.

Using Sentinel Cloud Portal, send each contact an email with credentials that enable the contact to log in to Sentinel Cloud Portal as a customer administrator and perform administrative tasks for users in their organization. For more information, see [Sentinel LDK Cloud Portal Guide for Customers](#).

8. The customer administrators log in to Sentinel Cloud Portal and define licensed users in their organization.

**NOTE** The vendor has the option of defining and managing licensed users directly in Sentinel LDK Cloud Portal without defining customer administrators.

9. The customer administrators can configure the client identity for each licensed user to allow access to one or more product licenses on the license server. Each client identity can also be configured with additional options depending on global parameters configured by the vendor.
10. Sentinel Cloud Portal sends each licensed user an email with a link that allows the user to easily install the identity string on their machine.

**NOTE** To use the provided link, Sentinel LDK Run-time Environment must be present on the user's machine. Sentinel Run-time Environment can be installed together with the protected application. If installing Sentinel LDK Run-time Environment is not possible, you can send the identity string to the user as a text string, with instruction on how to install the string manually. For details, see ["Installing a Client Identity on an End User's Machine" on the next page](#).

**11.** The end user installs the relevant protected application. The user can register their identity string on multiple machines (up to the limit you specify).

**12.** The end user proceeds to execute the protected application. The application consumes a network seat from the vendor's license server machine.

OR

The end user detaches a network seat from the license server machine to the local machine. The user can then proceed to execute the protected application offline.

## Setting Up the License Server Machine

This section describes how to set up your license server machine.

The license server machine is the repository for network licenses. The machine must be connected to the Internet and available at all times.

**NOTE** You should always update the license server machine with the latest Run-time Environment to ensure the best security and compatibility.

## Setting Up Admin Control Center

Install Admin Control Center on the license server machine. Configure the parameters described below in Admin Control Center.

### Limit Configuration Activities to an ACC Administrator

1. Go to **Configuration** page > **Basic Settings** tab.
2. For the **Password Protection** parameter, select **All ACC Pages**.
3. On the same line, click **Change Password**.
4. Enter the new password in the **New Admin Password** field and in the **Re-enter new Admin Password** field.
5. Click **Submit**. The new password is set.

## Configure Access From Remote Clients

1. Go to **Configuration** page > **Configuring Access From Remote Clients** tab.
2. Configure these parameters:
  - **Allow Access From Remote Clients.** If the license server machine will be used only to serve cloud licenses to machines with identity clients, select **Identifiable clients only**. Otherwise, select the relevant option.
  - **Public Address for Access With Identity**
  - **Store Identity Secrets.** Select the **Encrypted** option if you want identity secrets stored in the License Manager database to be encrypted. If you select this option, you must also specify the storage encryption key using Sentinel Admin API.

## Allow Detaching of Licenses

To allow end users to detach licenses, be sure to select the configuration parameter **Enable Detaching Licenses**.

## Setting Up Cloud Licenses

Cloud licenses for identity-based accesses must satisfy the following requirements:

- > Your Sentinel LDK Master license must contain the [Cloud Licensing module](#).
- > When using Sentinel LDK-EMS, the parameter **Cloud Licensing** in the Administration Console must be set to **Enabled**.
- > The license must have the SL AdminMode locking type and must be defined with concurrency, specifying the desired number of network seats to make available.

Set up separate CL keys for each customer. Use the **Produce and Push** option in Sentinel LDK-EMS to push the keys to the license server.

## Installing a Client Identity on an End User's Machine

Each client identity is represented by a unique identity string. You obtain the string for each client identity using Sentinel LDK Cloud Portal. The licensed user that receives the identity string can install it on their machines using one of the following methods:

- > Install the identity string using an email from Sentinel LDK Cloud Portal.
- > Enter the identity string manually in the License Manager configuration file on the machine.

Each of these methods is described in this section.



## Installing an Identity String Using an Email from Sentinel LDK Cloud Portal

Using Sentinel LDK Cloud Portal, the vendor administrator or the customer administrator can send an email to a licensed user that contains a link for installing the user's identity string on their machine. The user simply clicks the link to install the identity string.

This option can only be used if Sentinel LDK Run-time Environment 8.51 or later is installed on the user's machine.

## Entering an Identity String Manually

For a machine that does not have the Run-time Environment installed, the licensed user must enter the identity string manually in the configuration file for the Integrated License Manager or the External License Manager. (The vendor administrator or customer administrator uses Sentinel LDK Cloud Portal to generate an identity string for the licensed user.)

To determine where to find the configuration file for the License Manager, see ["Working Directly With License Manager Configuration Files" on page 221](#).

For a Windows desktop application, this file is typically:

```
%LocalAppData%\SafeNet Sentinel\Sentinel LDK\hasp_<vendorId>.ini
```

where *<vendorId>* is the Vendor ID associated with your Batch Code (or **hasp\_demo** for DEMOMA).

Open the configuration file in a text editor. Enter the identity string using the **serveraddr** parameter. For example:

```
serveraddr = 4VE5N3V:oBWAAQCBEBJnJ3btDmfhbLtCw/o7kjE@10.162.104.213
```

The configuration file can contain multiple **serveraddr** parameters.

If proxy communication is required, manually set proxy attributes in the configuration file described above.

Add the following parameters in the configuration file:

```
proxy = 1
proxy_host = <host>
proxy_port = 8080
proxy_username = <username>
proxy_password = <password>
```

Provide the required values where indicated.

## Cloud Licensing Performance

This section describes considerations for system performance when implementing cloud licensing.

- Sentinel LDK cloud licensing has been proven to support 100,000 concurrent login accesses from remote clients running an application that was protected with Sentinel LDK Envelope using the default settings. This is not an absolute limit; in most cases, a greater number of clients can be supported.

Performance may be affected by server resources, network latency, and client bandwidth.

The number of clients stated above applies equally to the following:

- LAN and cloud environments (This was tested with Amazon Web Services and Alibaba Cloud.)
  - Windows and Linux servers
  - All license models
- > The maximum number of SL keys with cloud licenses that can exist on a license server machine (Windows or Linux) is unlimited.

## Overview for Multi-Level License Servers

---

Cloud licensing gives you the option of allowing customers to set up a second level of license servers to provide greater control over the distribution of network seats within an organization.

### Working with Multi-Level License Servers

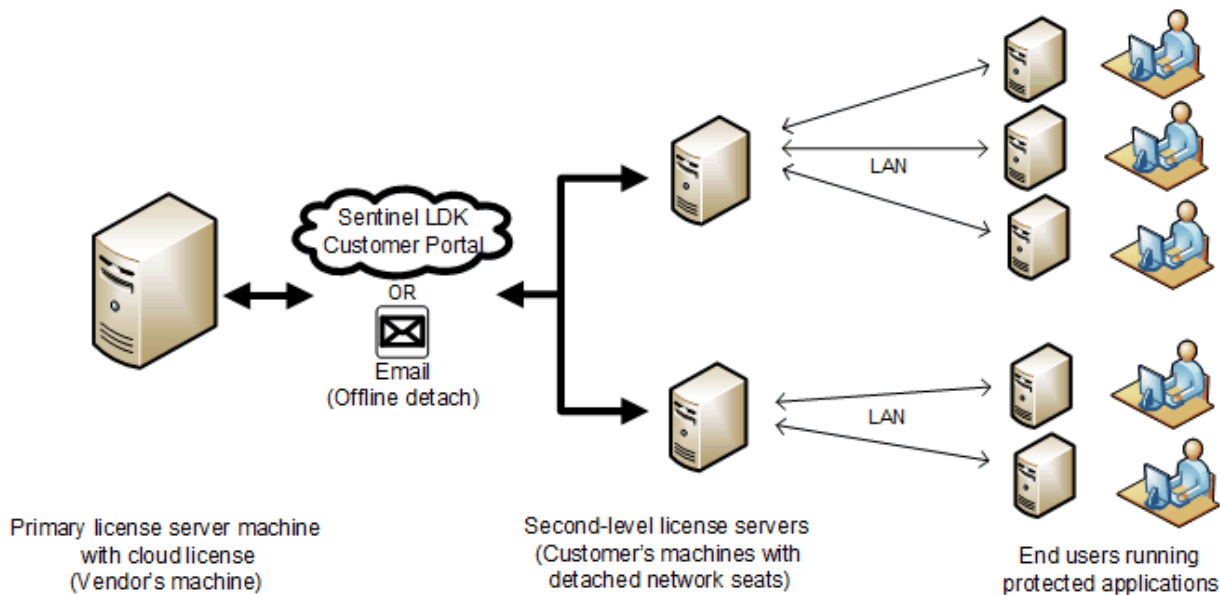
Multi-level license servers enable a customer to:

- > Control the distribution of network seats from a license with concurrency among users connected to LANs at different locations.
- > Ensure the availability of network seats at remote locations, even if Internet connectivity to the cloud license server is not available or is not reliable.
- > Allow groups of users to access network seats, without providing a client identity to each user.

For a given customer, the vendor defines the total number of network seats available to the customer. The customer decides how to divide the seats among their secondary license servers.

When an end user starts a protected application from a machine on the same LAN as a second-level license server, the protected application consumes a seat or automatically detaches a seat from the second-level license server.

The detached license on the second-level license server acts as an SL license, not a cloud license. The end user does not require a client identity.



The primary license server machine is the main repository for licenses. Licenses are detached from the primary license server machine to second-level license server machines either online or offline (as described below).

When working with multi-level license servers, the following limitations exist:

- > Network seats cannot be detached from a Product if all of the Features in the Product have unlimited concurrency.
- > A network seat can be detached from a second-level license server using automatic detach. However, it cannot be detached using on-demand detach.
- > A detached license with concurrent seats cannot be partially cancelled. If the license is cancelled, all the seats are cancelled.
- > As long as any network seats are detached (using automatic detach) from the second-level license server, the detached license on the server cannot be cancelled.

Cloud licensing with multi-level license servers can be implemented in the following configurations:

- > The vendor hosts the SL licenses on their own primary license server. The customer sets up one or more second-level license servers to host the detached network seats.

A license is detached from the primary license server using one of the following methods:

- **Online detach**

The vendor uses Sentinel LDK Cloud Portal to define a licensed user with special permission to detach network seats with concurrency for the customer. The customer uses the notification email generated by Sentinel LDK Cloud Portal to install the client identity on each second-level license server.

Using the second-level license server, the customer detaches a license with network seats from the SL key on the primary license server over the Internet. The customer has direct control over how many network seats from the pool are assigned to each second-level server.

- **Offline detach**

Using email or other offline means, the customer sends a machine ID file to the vendor for each second-level server to receive a detached license with network seats.

Using the primary license server, the vendor applies the machine ID files and detaches the required number of network seats from the SL key for each of the customer's second-level license servers. The vendor coordinates with the customer to determine how many seats are required for each second-level server. The vendor sends the required number of seats for each second-level license server to the customer in an H2R file by email or other offline means.

The customer applies each H2R file on the appropriate second-level server.

The customer does not require a client identity on the second-level server.

This section describes how to set up your license server machines.

## Implementation Summary

The following steps summarize the process of implementing the cloud licensing model with multi-level servers. Detailed instructions are provided in the sections that follow.

Each of the steps below is performed by the vendor unless stated otherwise:

1. Protect your applications using Sentinel LDK. Ensure that the **Locking Type** includes the option to use SL AdminMode keys.
2. Use Sentinel LDK-EMS to generate a separate CL license for each customer organization. Ensure that your Sentinel LDK Master license contains the required modules. The license must allow concurrency. Specify the total number of network seats that are available to the customer organization.
3. Set up a primary license server machine. Using **Produce & Push** in Sentinel LDK-EMS, install each SL license in a separate CL key on the server.

4. Do one of the following:

**To support online detach:**

- a. Ensure that the primary server machine has Internet access.
- b. For each customer organization that will use second-level license servers, use Sentinel LDK Cloud Portal to generate a special licensed user. Make sure that in the attributes for the user, the options **Allow Connection to Licenses**, **Allow License Detaching**, and **Allow Concurrency for Detached Licenses** are all set to **Yes**.
- c. The customer's administrator installs Sentinel Run-time Environment and uses the email notification from Sentinel LDK Cloud Portal to install the client identity on one or more machines that will serve as second-level license servers.

- d. For each second-level license server, the local administrator uses Admin Control Center to detach the required number of seats and sets up the License Manager to allow access to the relevant end users.

**To support offline detach:**

- a. The customer's local administrator installs Sentinel Run-time Environment on one or more machines that will serve as second-level license servers.
  - b. On each second-level machine, the local administrator generates a machine ID file and sends the file to you (the vendor).
  - c. Apply the machine ID files on the primary license server machine.
  - d. After consulting with the customer, detach the required number of seats for the machine ID of each second-level license server. An H2R file is generated for each server. Send the files to the customer.
  - e. For each second-level license server, the local administrator uses Admin Control Center to apply the provided H2R file to install the detached licenses. The local administrator sets up the License Manager to allow access for the relevant end users to the server.
5. The end users install and run Sentinel Admin Control Center and the protected application on a machine in the same LAN as the second-level license server. Each instance of the protected application consumes a network seat from the second-level license server.

## Online Detach: Setting Up Multi-Level License Server Machines

This section describes how to set up the primary and second-level license server machines when working with online detach of licenses.

The primary license server machine is the repository for network licenses. The machine only requires connection to the Internet when performing the following actions:

- > When detaching concurrent seats from the license server machine to the second-level license server.
- > When modifying the number of concurrent seats detached from the license server machine or when extending the date for the detached seats.
- > When canceling the detach operation.

**NOTE** Sentinel LDK Run-time Environment 8.51 or later must be installed on both the primary license server machine and on each second-level license server machine. You should always update the license server machines with the latest Run-time Environment to ensure the best security and compatibility.

## Setting Up the Primary License Server Machine

Set up the primary license server machine as described in this section.

### Set Up Admin Control Center

Install Admin Control Center on the primary license server machine. Configure the parameters as described in ["Setting Up the License Server Machine" on page 159](#).

Under **Allow Detaching of Licenses**, be sure to select the configuration parameter **Enable Detaching Licenses**.

### Set Up Cloud Licenses

To set up cloud licenses for identity-based access, see ["Setting Up Cloud Licenses" on page 160](#).

### Set Up a Client Identity for the Second-Level License Servers

On the primary license server machine, in the SL key for the customer organization, set up a dedicated licensed user to be installed on all the second-level license servers.

#### To set up the dedicated client identity:

1. Use Sentinel LDK Cloud Portal to define a special-purpose licensed user. Make sure that in the attributes for the user, the options **Allow Connection to Licenses**, **Allow License Detaching**, and **Allow Concurrency for Detached Licenses** are all set to **Yes**.
2. The notification email generated for the special-purpose licensed user should be sent to an administrator in the customer's organization who will ensure that the email is used to install the client identity on one or more machines that will serve as second-level license servers.

## Setting Up Second-Level License Servers

The customer organization sets up one or more second-level license servers on which the detached network seats will reside.

#### To set up a second-level license server machine:

1. Install the Sentinel LDK Run-time Environment on the machine.
2. In Admin Control Center, limit configuration activities to an ACC administrator as described earlier.
3. Use the email notification generated by Sentinel LDK Cloud Portal to install the identity string for the special-purpose licensed user generated above on the machine.
4. Using Admin Control Center, go to **Configuration > Users** and specify which end users can access seats on the second-level license server machine.

5. Using Admin Control Center, detach the required number of seats from the primary license server machine as follow:
  - a. Go to the Products page. Select the relevant Product and click the **Detach** button.
  - b. On the Detach License page, select the **Online** detach method.
  - c. On the **Concurrency** section of the page, select **Allow Concurrency for Detached License**.
  - d. In **Total Number of Seats**, enter the number of seats to detach.
  - e. On the **Specify Expiration Date for Detached License** section of the page, enter the expiration date for the detached seats.
  - f. Click **Detach & Attach**. The specified number of seats are detached from the license on the primary license server machine and are attached to the second-level license server machine.
  - g. (Optional) Set up the server machine to support automatic detach of seats. Go to **Configuration > Detachable Licenses**. Under **Automatic Detaching of Licenses**, select **Enabled** and assign a value for **Allowed Offline Duration**. Click **Submit**.

## Setting Up End Users

Each end user must be in the same LAN as the second-level license server. The Sentinel LDK Run-time Environment must be installed on each end user's machine. Each end user must have access to the protected application.

## Offline Detach: Setting Up Multi-Level License Server Machines

This section describes how to set up the primary and second-level license server machines when working with offline detach of licenses. The setup is the same for both the vendor-hosted implementation and the customer-hosted implementation, except as noted.

The primary license server machine is the repository for network licenses. The machine does not require connection to the Internet.

**NOTE** Sentinel LDK Run-time Environment 8.23 or later must be installed on both the primary license server machine and on each second-level license server machine. You should always update the license server machines with the latest Run-time Environment to ensure the best security and compatibility.

## Setting Up Second-Level License Servers

The customer organization sets up one or more second-level license servers on which the detached network seats will reside. Use the procedure that follows for each second-level license server machine.

### Set up Admin Control Center on the second-level license server machine

1. Install the Sentinel LDK Run-time Environment on the second-level license server machine. Perform the remainder of the steps in Admin Control Center.
2. Limit configuration activities to an ACC administrator as described earlier.
3. Go to **Configuration > Users** and specify which end users can access seats on the second-level license server machine.
4. Generate a machine ID file as follows:
  - a. Using Admin Control Center, go to **Diagnostics**.
  - b. Click the **Create ID File** button. Save the generated machine ID file.
  - c. Transfer the machine ID file to a location that is accessible from the primary license server machine.
5. (Optional) Set up the second-level license server machine to support automatic detach of seats:
  - a. Go to **Configuration > Detachable Licenses**.
  - b. Under **Automatic Detaching of Licenses**, select **Enabled** and assign a value for **Allowed Offline Duration**.
  - c. Click **Submit**.

## Setting Up the Primary License Server Machine

Set up the primary license server machine as described in this section.

### Set Up Cloud Licenses

To set up cloud licenses for identity-based access, see .

### Set Up Admin Control Center on the Primary License Server Machine

1. Install the Sentinel LDK Run-time Environment on the primary license server machine. Perform the remainder of the steps in Admin Control Center.
2. Configure the parameters as described in .
3. Under **Allow Detaching of Licenses**, be sure to select the configuration parameter **Enable Detaching Licenses**.
4. For each second-level license server machine, apply the machine ID file that was prepared on that machine as follows:



- a. Go to **Update/Attach**.
  - b. Click **Select File** and select the relevant machine ID file.
  - c. Click **Apply File**.
5. For each second-level license server machine, detach the required number of seats from the primary license server machine as follow:
  - a. Go to the Products page. Select the relevant Product and click the **Detach** button.
  - b. On the Detach License page, select the **Offline** detach method. Under **Select Recipient Machine**, select the ID of the target second-level license server machine.
  - c. On the **Concurrency** section of the page, select **Allow Concurrency for Detached License**.
  - d. In **Total Number of Seats**, enter the number of seats to detach for the selected license server machine.
  - e. On the **Specify Expiration Date for Detached License** section of the page, enter the expiration date for the detached seats.
  - f. Click **Detach**. An H2R file that contains the detached license is generated.
  - g. Transfer the H2R file to a location that is accessible from the relevant second-level license server machine.

## Attaching the License to the Second-Level License Server

The customer organization attaches the license that was detached from the primary license server to the second-level license servers on which the detached network seats will reside.

### To attach the license to the second-level license server machine:

1. In Admin Control Center, go to **Update/Attach**.
2. Click **Select File**. Select the relevant H2R file transferred from the primary license server machine.
3. Click **Apply File**.

## Setting Up End Users

Each end user must be in the same LAN as the second-level license server. The Sentinel LDK Run-time Environment must be installed on each end user's machine. Each end user must have access to the protected application.

## Configuring High Availability for Cloud Licensing

---

Sentinel LDK supports configuring a vendor-hosted cloud license server for high availability.

Sentinel LDK License Managers in the vendor's data center can be configured to store licenses in a common external trusted license storage (a MySQL database cluster).

You can set up License Managers on two license server machines (active and passive). If the active License Manager stops responding, a load balancer will handle failover from the active License Manager to the passive one. Only one License Manager will serve licenses at any point in time.

For information on configuring high availability, see the [Sentinel LDK High Availability for Cloud Licensing Configuration Guide](#).

# CHAPTER 16: Cloud Licensing Using Sentinel Admin Control Center

This section describes how to implement cloud licensing in Sentinel LDK when using Sentinel Admin Control Center to manage licensed users.

**NOTE** To implement cloud licensing using Sentinel LDK Cloud Portal, see ["Cloud Licensing Using Sentinel LDK Cloud Portal" on page 156](#).

*In this section:*

- > ["Cloud Licensing Overview" below](#)
- > ["Vendor-Hosted Implementation Summary" on page 173](#)
- > ["Setting Up the License Server Machine" on page 174](#)
- > ["Installing a Client Identity on an End User's Machine" on page 176](#)
- > ["Customer-Hosted Implementation Summary" on page 178](#)
- > ["Cloud Licensing Performance" on page 178](#)
- > ["Overview for Multi-Level License Servers" on page 179](#)
- > ["Online Detach: Setting Up Multi-Level License Server Machines " on page 182](#)
- > ["Offline Detach: Setting Up Multi-Level License Server Machines" on page 184](#)
- > ["Configuring High Availability for Cloud Licensing" on page 187](#)

## Cloud Licensing Overview

Cloud licensing enables end users to access local software with a license hosted in the cloud.

Cloud licensing uses identity-based access to give you granular control over who can access a network seat from an SL license. Cloud licensing can be implemented at either of two levels:

- > A software vendor can set up a license server machine and grant access to cloud licenses over the Internet for their customers.

- > A customer who has purchased an SL license can set up a license server machine and grant access to cloud licenses for authorized end user within their organization. Access can be granted either over the Internet or over the company network (or both).

The benefits provided for each level are described below.

- > **Vendor-hosted implementation:** Cloud licensing provides an alternative model for distributing licenses for protected applications to customers. Rather than providing SL protection keys to the customer, you host the SL licenses on your own license server and provides *client identities* for each end user. Each client identity is represented by a unique *identity string* that the end user installs on their machine.

The client identity entitles the user to execute an instance of the protected application, detach a network seat that can be used offline, or both. The user can install the identity string on multiple machines (if permitted by the vendor), but can only execute a single instance or detach a single network seat at any point in time.

Among the benefits of using client identities are the following:

- Software can easily be used on different machines using a single identity.
- Client identities are not affected by operating system upgrades or hardware upgrades.
- You can easily disable a client identity if necessary.

Client identities can be assigned an expiration date. This provides the following additional benefits:

- You can easily manage trial by providing users with expiring identities that consume the same license. This eliminates the need to generate a new key and a new license for each trial.
- You can easily provide customers with an emergency cloud license in case they face issues with their local HL or SL key.
- The IT administrator can manage which users can access a concurrency license and for how long.

Client identities are managed using Sentinel Admin Control Center. You use the **Client Identities** tab of the Configuration page to create and manage client identities. For more information, see the [Sentinel Admin Control Center help](#).

For more information on the benefits of cloud licensing, see "[Understanding Cloud Licensing](#)" on page 33.

- > **Customer-hosted implementation:** You can enable your customers to implement cloud licensing at their site to control access to network seats from SL licenses on their license server machine. This provides a simplified method of controlling which end users are able to consume network seats from a license.

**NOTE** For either level of implementation, cloud licensing is only available if you have the [Cloud Licensing module](#) in your Sentinel LDK Master license at the time that you generate the SL licenses.

Client identities are managed using Sentinel Admin Control Center. Your customer uses the **Client Identities** tab of the Configuration page to create and manage client identities. For more information, see the [Sentinel Admin Control Center help](#).

Many of the benefits of working with client identities described above for the vendor-hosted implementation are also applicable for the customer-hosted implementation.

**For both vendor-hosted and customer-hosted implementations:** You also have the option of allowing customers to set up a second level of license servers to provide greater control over the distribution of network seats within an organization and to minimize the overhead of license administration. For more information, see ["Overview for Multi-Level License Servers" on page 179](#).

## Vendor-Hosted Implementation Summary

The following steps summarize the process of implementing the cloud licensing model. More details are provided in the sections that follow.

1. Protect your applications using Sentinel LDK. Ensure that the **Locking Type** includes the option to use SL AdminMode keys.
2. Use Sentinel LDK-EMS or Sentinel License Generation API to generate a separate SL AdminMode license for each customer organization. Ensure that your Sentinel LDK Master license contains the required modules. If you are using License Generation API, ensure that the license definition includes the tag **<cloud\_licensing>** with the value **Yes**. The license must allow concurrency. Specify the required number of network seats.
3. Set up a global license server machine that allows access via the Internet, and install each SL license in a separate SL key on the server.
4. For each unique end user, use Sentinel Admin Control Center to generate a client identity. Each client identity can be configured to allow access to a specific SL key on the license server. Each client identity is also configured whether to:
  - Allow the user's machine to log in to a license on the SL key.
  - Allow the user to detach a network seat from the key.
  - Allow the user to install the client identity on multiple machines. (Only one of the machines can consume a network seat at any given moment.) If you limit the number of machines, the License Manager automatically registers each machine when first used and allows you to view and manage the list. If you allow unlimited machines, the machines are not tracked.

For more information, see the **Configuration** screen – **Client Identities** tab in Admin Control Center.

5. Copy the identity string from the client identity and provide it to the end user.
6. The end user installs the relevant protected application. The user registers their identity string on their machine using either Admin Control Center or a text editor. The user can register their identity string on multiple machines (up to the limit you specify).
7. The end user proceeds to execute the protected application. The application consumes a network seat from the vendor's license server machine.

OR

The end user detaches a network seat from the license server machine to the local machine. The user can then proceed to execute the protected application offline.

## Setting Up the License Server Machine

This section describes how to set up your license server machine. The setup is the same for both the vendor-hosted implementation and the customer-hosted implementation, except as noted.

The license server machine is the repository for network licenses. The machine must be connected to the Internet and available at all times. (For customer-hosted implementation, connection to the Internet is optional.)

**NOTE** You should always update the license server machine with the latest Run-time Environment to ensure the best security and compatibility.

## Setting Up Admin Control Center

Install Admin Control Center on the license server machine. Configure the parameters described below in Admin Control Center.

### Limit Configuration Activities to an ACC Administrator

1. Go to **Configuration** page > **Basic Settings** tab.
2. For the **Password Protection** parameter, select **All ACC Pages**.
3. On the same line, click **Change Password**.
4. Enter the new password in the **New Admin Password** field and in the **Re-enter new Admin Password** field.
5. Click **Submit**. The new password is set.

### Configure Access From Remote Clients

1. Go to **Configuration** page > **Configuring Access From Remote Clients** tab.
2. Configure these parameters:
  - **Allow Access From Remote Clients**. If the license server machine will be used only to serve cloud licenses to machines with identity clients, select **Identifiable clients only**. Otherwise, select the relevant option.
  - **Public Address for Access With Identity**
  - **Store Identity Secrets**. Select the **Encrypted** option if you want identity secrets stored in the License Manager database to be encrypted. If you select this option, you must also specify the storage encryption key using Sentinel Admin API.

## Allow Detaching of Licenses

To allow end users to detach licenses, be sure to select the configuration parameter **Enable Detaching Licenses**.

## Setting Up Cloud Licenses

Cloud licenses for identity-based accesses must satisfy the following requirements:

- > Your Sentinel LDK Master license must contain the [Cloud Licensing module](#).
- > When using Sentinel LDK-EMS, the parameter **Cloud Licensing** in the Administration Console must be set to **Enabled**.
- > When using Sentinel License Generation API, the license definition must contain the tag: **<cloud\_licensing>Yes</cloud\_licensing>**
- > The license must have the SL AdminMode locking type and must be defined with concurrency, specifying the desired number of network seats to make available.

When you define a client identity, you can specify that it is valid only for a specific key ID. If you do not specify a key ID, the client identity is valid for all cloud licenses on the license server machine.

**For vendor-hosted implementation:** Set up separate SL keys for each customer.

- > If the customer is a single user, Thales recommends that you create a separate SL key for the user and link this key to the user's client identity.
- > If the end customer is an organization, Thales recommends that you create a single SL key and link all client identities in the organization to the key (assuming all users are all entitled to consume the same licenses). If various groups within the organization are entitled to use different sets of licenses, multiple keys can be used to partition the licenses and control access.

By default, the License Manager installs all SL licenses in the same SL key on a given machine. You can use the following procedure to install SL licenses in separate SL keys on the license server machine.

### To install multiple SL keys on the license server machine:

1. Ensure that no SL licenses are currently installed on the machine.
2. Use Admin Control Center or RUS to obtain a fingerprint of the machine. Save the fingerprint file for future use.
3. In Sentinel LDK-EMS, create a product key for a specific customer (or end user). Use the fingerprint file from step 2 to generate a V2C file for offline installation.
4. Use Admin Control Center or RUS to apply the V2C file on the license server machine. A new SL key (containing the new SL license) for the customer is created.
5. For each additional customer, repeat steps 3 and 4 as necessary to create new SL keys.

When you generate client identities for customers, you can specify the relevant key ID for each client.

This method allows you precise control over the licenses that you make available for each customer.

To generate and manage client identities for customers or end users, see the [Sentinel Admin Control Center help](#).

## Installing a Client Identity on an End User's Machine

Each client identity is represented by a unique identity string. You obtain the string for each client identity using Admin Control Center (go to **Configuration** page > **Configuring Client Identities** tab). The client (person or entity) that receives the identity string can install it on their machines using one of the following methods:

- > Install the identity string using Admin Control Center on the machine.
- > Enter the identity string manually in the License Manager configuration file on the machine.

Each of these methods is described in this section.

### Identity String Syntax

The identity string generated by Admin Control Center uses the following syntax:

```
<identityString>@<address>
```

where:

<identityString>	A unique string generated for each client identity.
<address>	The license server machine address. This address is taken from the field <b>Public Address for Access With Identity</b> that is specified in Admin Control Center as described earlier (see <a href="#">"Configure Access From Remote Clients" on page 174</a> ).

For example:

```
4VE5N3V:0BWAAQCBEbJnJ3btDmfhhLtCw/o7kjE@10.162.104.213
```

### Installing an Identity String Using Admin Control Center

The user can use Admin Control Center on the remote machine to install the identity string.

#### To install the identity string using Admin Control Center:

1. In Admin Control Center on the remote machine, go to **Configuration** page > **Access to Remote License Managers** tab (or enter this URL in a browser on the remote machine: `http://localhost:1947/_int_/config_to.html`). The **Configuration for Sentinel License Manager** tab is displayed.
2. Ensure that **Allow Access to Remote Licenses** is selected.



3. In **Remote License Search Parameters**, enter the identity string using one of the formats described above. The screen should now look similar to the following:

Sentinel Admin Control Center

Configuration Host Name: ndsm10033570-03

Sentinel Keys

Products

Features

Sessions

Update/Attach

Access Log

Configuration

Diagnostics

Basic Settings Users Access to Remote License Managers Access from Remote Clients Client Identities Detachable Licenses Network

Allow Access to Remote Licenses ☒ You may experience a delay of a few minutes before your changes take effect.

Broadcast Search for Remote Licenses ☐

Aggressive Search for Remote Licenses ☐

Remote License Search Parameters

4VE5N3V:oBWAAQCBEJnJ3btDmfbbLtCw/o7kjE@10.162.104.213

Submit Cancel Set Defaults

4. Click **Submit**.
5. If proxy communication is required, go to **Configuration** page > **Network** tab. Enter the required proxy settings and click **Submit**.

## Entering an Identity String Manually

For a remote machine that does not have the Run-time Environment installed, the user must enter the identity string manually in the configuration file for the Integrated License Manager or the External License Manager.

To determine where to find the configuration file for the License Manager, see ["Working Directly With License Manager Configuration Files" on page 221](#).

For a Windows desktop application, this file is typically:

```
%LocalAppData%\SafeNet Sentinel\Sentinel LDK\hasp_<vendorId>.ini
```

where <vendorId> is the Vendor ID associated with your Batch Code (or **hasp\_demo** for DEMOMA).

Open the configuration file in a text editor. Enter the identity string using the **serveraddr** parameter. For example:

```
serveraddr = 4VE5N3V:oBWAAQCBEJnJ3btDmfbbLtCw/o7kjE@10.162.104.213
```

The configuration file can contain multiple **serveraddr** parameters.

If proxy communication is required, manually set proxy attributes in the configuration file described above.

Add the following parameters in the configuration file:

```
proxy = 1
proxy_host = <host>
proxy_port = 8080
proxy_username = <username>
proxy_password = <password>
```

Provide the required values where indicated.

## Customer-Hosted Implementation Summary

The following steps summarize the actions that you must perform in order to enable your customer to implement the cloud licensing model.

1. Protect your applications using Sentinel LDK. Ensure that the **Locking Type** includes the option to use SL AdminMode keys.
2. Use Sentinel LDK-EMS or Sentinel License Generation API to generate an SL AdminMode license for each customer organization. Ensure that your Sentinel LDK Master license contains the required modules. If you are using License Generation API, ensure that the license definition includes the tag **<cloud\_licensing>** with the value **Yes**. The license must allow concurrency. Specify the required number of network seats.

The customer will perform the remainder of the required steps as documented in the [Sentinel Admin Control Center help](#).

## Cloud Licensing Performance

This section describes considerations for system performance when implementing cloud licensing.

- > Sentinel LDK cloud licensing has been proven to support 100,000 concurrent login accesses from remote clients running an application that was protected with Sentinel LDK Envelope using the default settings. This is not an absolute limit; in most cases, a greater number of clients can be supported.

Performance may be affected by server resources, network latency, and client bandwidth.

The number of clients stated above applies equally to the following:

- LAN and cloud environments (This was tested with Amazon Web Services and Alibaba Cloud.)
- Windows and Linux servers
- All license models

- > The maximum number of SL keys with cloud licenses that can exist on a license server machine (Windows or Linux) is unlimited.

## Overview for Multi-Level License Servers

---

Cloud licensing gives you the option of allowing customers to set up a second level of license servers to provide greater control over the distribution of network seats within an organization and to minimize the overhead of license administration.

### Working with Multi-Level License Servers

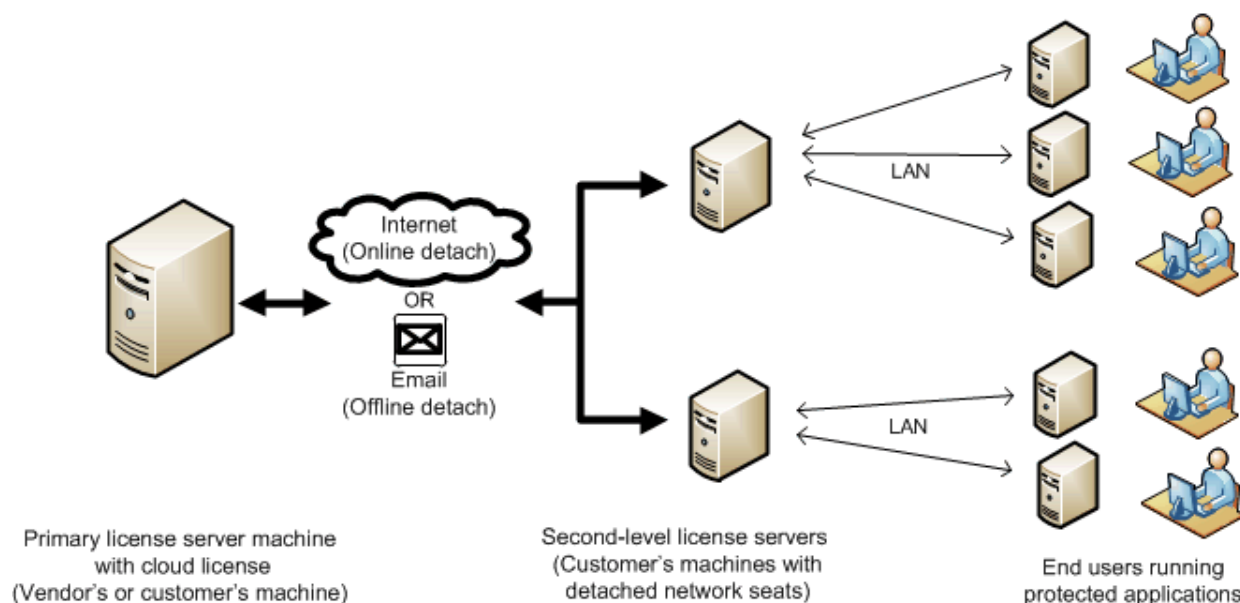
Multi-level license servers enable a customer to:

- > Control the distribution of network seats from a license with concurrency among users connected to LANs at different locations.
- > Ensure the availability of network seats at remote locations, even if Internet connectivity to the cloud license server is not available or is not reliable.
- > Allow groups of users to access network seats without providing a client identity to each user.

For a given customer, the vendor defines the total number of network seats available to the customer. The customer decides how to divide the seats among their secondary license servers.

When an end user starts a protected application from a machine on the same LAN as a second-level license server, the protected application consumes a seat or automatically detaches a seat from the second-level license server.

The detached license on the second-level license server acts as an SL license, not a cloud license. The end user does not require a client identity.



The primary license server machine is the main repository for licenses. Licenses are detached from the primary license server machine to second-level license server machines either online or offline (as described below).

When working with multi-level license servers, the following limitations exist:

- > Network seats cannot be detached from a Product if all of the Features in the Product have unlimited concurrency.
- > A network seat can be detached from a second-level license server using automatic detach. However, it cannot be detached using on-demand detach.
- > A detached license with concurrent seats cannot be partially cancelled. If the license is cancelled, all the seats are cancelled.
- > As long as any network seats are detached (using automatic detach) from the second-level license server, the detached license on the server cannot be cancelled.

Cloud licensing with multi-level license servers can be implemented in the following configurations:

- > **Vendor-hosted implementation:** The vendor hosts the SL licenses on their own primary license server. The customer sets up one or more second-level license servers to host the detached network seats.

A license is detached from the primary license server using one of the following methods:

- **Online detach**

The vendor defines a client identity with special permission to detach network seats with concurrency for the customer. The customer installs the client identity on each second-level license server.

Using the second-level license server, the customer detaches a license with network seats from the SL key on the primary license server over the Internet. The customer has direct control over how many network seats from the pool are assigned to each second-level server.

- **Offline detach**

Using email or other offline means, the customer sends a machine ID file to the vendor for each second-level server to receive a detached license with network seats.

Using the primary license server, the vendor applies the machine ID files and detaches the required number of network seats from the SL key for each of the customer's second-level license servers. The vendor coordinates with the customer to determine how many seats are required for each second-level server. The vendor sends the required number of seats for each second-level license server to the customer in an H2R file by email or other offline means.

The customer applies each H2R file on the appropriate second-level server.

The customer does not require a client identity on the second-level server.

- **Customer-hosted implementation:** You can enable your customers to host the SL licenses on their own primary license server and to set up second-level license servers to host the detached concurrent network seats.

The customer can use **online detach** or **offline detach** as described above. However, for the customer-hosted implementation, all of the license servers are controlled by the customer.

**NOTE** For either level of implementation, cloud licensing is only available if you have the Cloud Licensing module in your Sentinel LDK Master license at the time that you generate the SL licenses.

This section describes how to set up your license server machines. The setup is the same for both the vendor-hosted implementation and the customer-hosted implementation, except as noted.

## Implementation Summary

The following steps summarize the process of implementing the cloud licensing model with multi-level servers. Detailed instructions are provided in the sections that follow.

Each of the steps below is performed by the vendor unless stated otherwise:

1. Protect your applications using Sentinel LDK. Ensure that the **Locking Type** includes the option to use SL AdminMode keys.
2. Use Sentinel LDK-EMS or Sentinel License Generation API to generate a separate SL AdminMode license for each customer organization. Ensure that your Sentinel LDK Master license contains the required modules. If you are using License Generation API, ensure that the license definition includes the tag **<cloud\_licensing>** with the value **Yes**. The license must allow concurrency. Specify the total number of network seats that are available to the customer organization.
3. Set up a primary license server machine. Install each SL license on the server. For vendor-hosted implementation, install each SL license in a separate SL key on the server.
4. Do one of the following:

**To support online detach:**

- a. Ensure that the primary server machine has Internet access.
- b. For each customer organization that will use second-level license servers, use Sentinel Admin Control Center on the primary license server machine to generate a client identity. Configure the client identity to allow detaching a license with concurrent seats (select **Allow Concurrency for Detached Licenses**).  
For more information, see the **Configuration** screen – **Client Identities** tab in Admin Control Center.
- c. Using Sentinel Admin Control Center, copy the identity string from the client identity. Provide the identity string to the customer organization.
- d. The customer's administrator installs Sentinel Run-time Environment and installs the client identity on one or more machines that will serve as second-level license servers.
- e. For each second-level license server, the local administrator uses Admin Control Center to detach the required number of seats and sets up the License Manager to allow access to the relevant end users.

**To support offline detach:**

- a. The customer's local administrator installs Sentinel Run-time Environment on one or more machines that will serve as second-level license servers.
  - b. On each second-level machine, the local administrator generates a machine ID file and sends the file to you (the vendor).
  - c. Apply the machine ID files on the primary license server machine.
  - d. After consulting with the customer, detach the required number of seats for the machine ID of each second-level license server. An H2R file is generated for each server. Send the files to the customer.
  - e. For each second-level license server, the local administrator uses Admin Control Center to apply the provided H2R file to install the detached licenses. The local administrator sets up the License Manager to allow access for the relevant end users to the server.
5. The end users install and run Sentinel Admin Control Center and the protected application on a machine in the same LAN as the second-level license server. Each instance of the protected application consumes a network seat from the second-level license server.

## Online Detach: Setting Up Multi-Level License Server Machines

This section describes how to set up the primary and second-level license server machines when working with online detach of licenses. The setup is the same for both the vendor-hosted implementation and the customer-hosted implementation, except as noted.

The primary license server machine is the repository for network licenses. The machine only requires connection to the Internet when performing the following actions:

- > When detaching concurrent seats from the license server machine to the second-level license server.
- > When modifying the number of concurrent seats detached from the license server machine or when extending the date for the detached seats.
- > When canceling the detach operation.

**NOTE** Sentinel LDK Run-time Environment 8.23 or later must be installed on both the primary license server machine and on each second-level license server machine. You should always update the license server machines with the latest Run-time Environment to ensure the best security and compatibility.

## Setting Up the Primary License Server Machine

Set up the primary license server machine as described in this section.

## Set Up Admin Control Center

Install Admin Control Center on the primary license server machine. Configure the parameters as described in ["Setting Up the License Server Machine" on page 174](#).

Under **Allow Detaching of Licenses**, be sure to select the configuration parameter **Enable Detaching Licenses**.

## Set Up Cloud Licenses

To set up cloud licenses for identity-based access, see ["Setting Up Cloud Licenses" on page 175](#).

## Set Up a Client Identity for the Second-Level License Servers

On the primary license server machine, in the SL key for the customer organization, set up a dedicated client identity to be installed on all the second-level license servers.

### To set up the dedicated client identity:

1. In Admin Control Center, go to the **Configuration > Client Identities > Add Client Identity** page. Add a client identity in which you select the parameters **Allow Remote Detach Access** and **Allow Concurrency for Detached Licenses**.
2. Provide the identity string to the customer organization.

## Setting Up Second-Level License Servers

The customer organization sets up one or more second-level license servers on which the detached network seats will reside.

### To set up a second-level license server machine:

1. Install the Sentinel LDK Run-time Environment on the machine.
2. In Admin Control Center, limit configuration activities to an ACC administrator as described [earlier](#).
3. Install the dedicated client identity generated above on the machine. For more information, see ["Installing a Client Identity on an End User's Machine" on page 176](#).

4. Using Admin Control Center, go to **Configuration > Users** and specify which end users can access seats on the second-level license server machine.
5. Using Admin Control Center, detach the required number of seats from the primary license server machine as follow:
  - a. Go to the Products page. Select the relevant Product and click the **Detach** button.
  - b. On the Detach License page, select the **Online** detach method.
  - c. On the **Concurrency** section of the page, select **Allow Concurrency for Detached License**.
  - d. In **Total Number of Seats**, enter the number of seats to detach.
  - e. On the **Specify Expiration Date for Detached License** section of the page, enter the expiration date for the detached seats.
  - f. Click **Detach & Attach**. The specified number of seats are detached from the license on the primary license server machine and are attached to the second-level license server machine.
  - g. (Optional) Set up the server machine to support automatic detach of seats. Go to **Configuration > Detachable Licenses**. Under **Automatic Detaching of Licenses**, select **Enabled** and assign a value for **Allowed Offline Duration**. Click **Submit**.

## Setting Up End Users

Each end user must be in the same LAN as the second-level license server. The Sentinel LDK Run-time Environment must be installed on each end user's machine. Each end user must have access to the protected application.

## Offline Detach: Setting Up Multi-Level License Server Machines

This section describes how to set up the primary and second-level license server machines when working with offline detach of licenses. The setup is the same for both the vendor-hosted implementation and the customer-hosted implementation, except as noted.

The primary license server machine is the repository for network licenses. The machine does not require connection to the Internet.

**NOTE** Sentinel LDK Run-time Environment 8.23 or later must be installed on both the primary license server machine and on each second-level license server machine. You should always update the license server machines with the latest Run-time Environment to ensure the best security and compatibility.



## Setting Up Second-Level License Servers

The customer organization sets up one or more second-level license servers on which the detached network seats will reside. Use the procedure that follows for each second-level license server machine.

### Set up Admin Control Center on the second-level license server machine

1. Install the Sentinel LDK Run-time Environment on the second-level license server machine. Perform the remainder of the steps in Admin Control Center.
2. Limit configuration activities to an ACC administrator as described [earlier](#).
3. Go to **Configuration > Users** and specify which end users can access seats on the second-level license server machine.
4. Generate a machine ID file as follows:
  - a. Using Admin Control Center, go to **Diagnostics**.
  - b. Click the **Create ID File** button. Save the generated machine ID file.
  - c. Transfer the machine ID file to a location that is accessible from the primary license server machine.
5. (Optional) Set up the second-level license server machine to support automatic detach of seats:
  - a. Go to **Configuration > Detachable Licenses**.
  - b. Under **Automatic Detaching of Licenses**, select **Enabled** and assign a value for **Allowed Offline Duration**.
  - c. Click **Submit**.

## Setting Up the Primary License Server Machine

Set up the primary license server machine as described in this section.

### Set Up Cloud Licenses

To set up cloud licenses for identity-based access, see ["Setting Up Cloud Licenses" on page 175](#).

### Set Up Admin Control Center on the Primary License Server Machine

1. Install the Sentinel LDK Run-time Environment on the primary license server machine. Perform the remainder of the steps in Admin Control Center.
2. Configure the parameters as described in ["Setting Up the License Server Machine" on page 174](#).
3. Under **Allow Detaching of Licenses**, be sure to select the configuration parameter **Enable Detaching Licenses**.
4. For each second-level license server machine, apply the machine ID file that was prepared on that machine as follows:

- a. Go to **Update/Attach**.
  - b. Click **Select File** and select the relevant machine ID file.
  - c. Click **Apply File**.
5. For each second-level license server machine, detach the required number of seats from the primary license server machine as follow:
  - a. Go to the Products page. Select the relevant Product and click the **Detach** button.
  - b. On the Detach License page, select the **Offline** detach method. Under **Select Recipient Machine**, select the ID of the target second-level license server machine.
  - c. On the **Concurrency** section of the page, select **Allow Concurrency for Detached License**.
  - d. In **Total Number of Seats**, enter the number of seats to detach for the selected license server machine.
  - e. On the **Specify Expiration Date for Detached License** section of the page, enter the expiration date for the detached seats.
  - f. Click **Detach**. An H2R file that contains the detached license is generated.
  - g. Transfer the H2R file to a location that is accessible from the relevant second-level license server machine.

## Attaching the License to the Second-Level License Server

The customer organization attaches the license that was detached from the primary license server to the second-level license servers on which the detached network seats will reside.

### To attach the license to the second-level license server machine:

1. In Admin Control Center, go to **Update/Attach**.
2. Click **Select File**. Select the relevant H2R file transferred from the primary license server machine.
3. Click **Apply File**.

## Setting Up End Users

Each end user must be in the same LAN as the second-level license server. The Sentinel LDK Run-time Environment must be installed on each end user's machine. Each end user must have access to the protected application.

## Configuring High Availability for Cloud Licensing

---

Sentinel LDK supports configuring a vendor-hosted cloud license server for high availability.

Sentinel LDK License Managers in the vendor's data center can be configured to store licenses in a common external trusted license storage (a MySQL database cluster).

You can set up License Managers on two license server machines (active and passive). If the active License Manager stops responding, a load balancer will handle failover from the active License Manager to the passive one. Only one License Manager will serve licenses at any point in time.

For information on configuring high availability, see the [Sentinel LDK Installation Guide](#).

## Part 4 - Distributing Software

### In this section:

---

- > ["Distributing Sentinel LDK With Your Software" on page 189](#) – Describes options for distributing required software to your end users.
- > ["Sentinel License Manager" on page 208](#) – Describes the facilities used by Sentinel LDK to manage access to local and remote Sentinel protection keys from a protected application.
- > ["Sentinel Admin Control Center" on page 229](#) – Describes the configuration and management functionality of Sentinel Admin Control Center, an end-user utility that enables centralized administration of Sentinel License Managers and Sentinel protection keys.

# CHAPTER 17: Distributing Sentinel LDK With Your Software

This section introduces options for distributing required software to your end users.

*In this section:*

- > ["Sentinel LDK Software for End Users" below](#)
- > ["Distributing Sentinel LDK Run-time Environment" on page 193](#)

## Sentinel LDK Software for End Users

Every Sentinel LDK installation includes software that you need to distribute to your end users. This software must be installed at your customer's site to ensure that your protected and licensed software functions correctly.

### Sentinel LDK Run-time Environment

In many instances, the Sentinel LDK Run-time Environment must be installed on the computer of each end user who will use the protected application so that the application can communicate with the Sentinel protection key. For network licenses, the Run-time Environment is required on the computer where the network license is located. For information on when the Run-time Environment is required, see ["Situations That Require Sentinel LDK Run-time Environment" on page 194](#).

There are a number of ways in which the Run-time Environment can be installed. For more information, see ["Distributing Sentinel LDK Run-time Environment" on page 193](#).

### Software for Sentinel HL Licenses

No Sentinel LDK software is required specifically for protected applications that are licensed with Sentinel HL (Driverless configuration) keys. However, other Sentinel LDK software may be required as described in this section.

Other HL keys may require the Run-time Environment. For more information, see ["Situations That Require Sentinel LDK Run-time Environment" on page 194](#).

### Software for Protected Applications Under Windows

Thales recommends that, under Windows, you include the External License Manager with all protected applications. For more information, see ["Types of License Managers" on page 209](#).

## Software for Sentinel HL Licenses Under Linux Intel

For an application licensed with a standalone Sentinel HL (Driverless configuration) key, you can optionally install the Sentinel LDK Run-time Environment installer for Linux on the machine where the protected application executes.

- > If you install the Run-time Environment, all required files are already deployed. For the list of files installed by the Run-time Environment installer, see the [Sentinel LDK Installation Guide](#).
- > If you do not install the Run-time Environment, you must manually copy the file `/Linux/Redistribute/Runtimeless/80-hasp.rules` into the `/etc/udev/rules.d` on the machine where the protected application executes. This sets the permissions to allow any user to access the Sentinel HL Driverless USB device. (Under Linux, by default only superusers have access to USB devices.)

**NOTE** An administrator account is required to copy this file.

## Software for Protected .NET Applications Under Linux Intel

To support the Envelope runtime for .NET applications:

- > Copy the complete protected application folder (including Linux runtime file: `libhaspdnert_x86_64.so`) to the target machine.
- > Copy the LDK Run-time API libraries (`libhasp_linux_x86_64_demo.so`) to one of the following directories:
  - To one of the directories specified in the `LD_LIBRARY_PATH` variable.
  - To `/var/hasplm`, `/usr/local/lib`, or `/usr/lib64`.

## Software for Sentinel HL Licenses Under Linux ARM

For an application that is licensed with a standalone Sentinel HL (Driverless configuration) key: Copy the file `/Linux/Redistribute/Runtimeless/80-hasp.rules` into the `/etc/udev/rules.d` directory on the machine where the protected application executes. This operation is normally performed by the Run-time Environment installer for Linux. However, if the Run-time Environment is not installed, the file must be copied manually.

**NOTE** If the Linux distribution supports **udev**, and the `80-hasp.rules` file is copied as described above, Sentinel Licensing API uses **udev** to access USB devices. Otherwise, the Licensing API uses **sysfs**.

**udev** allows both root and regular users to access USB devices. **sysfs** allows only root to access USB devices.

However, in some Linux ARM distributions, **udev** is configured to block regular users from accessing USB devices even though the `80-hasp.rules` file requests such access for these users. To bypass this limitation, the user can be added to the **plugdev** group of users. Users in this group are allowed access to USB devices.

## Software for Sentinel SL Licenses

When you distribute protected applications that are licensed with Sentinel SL keys, you must include the following Sentinel LDK components with your applications.

### For Unlocked Products

As part of the process to create unlocked protected applications, you create a bundle that includes your protected application and all the required Sentinel LDK components. Therefore, no additional components are required when you distribute the bundle.

Unlocked protected applications can also be updated to locked protected applications at the customer's site without adding any Sentinel LDK components.

### For Locked Products

When you distribute a locked protected application that is not installed first as an unlocked application, your installation procedure should also install your customized Vendor library (the **haspvlib\_vendorID.\*** file) in the following location:

- > **For Windows x64:** `%CommonProgramFiles(x86)%\Aladdin Shared\HASP\`
- > **For Windows x86:** `%CommonProgramFiles%\Aladdin Shared\HASP\`
- > **For Mac:** `/var/hasplm` (By default, the `/var` path is hidden. You may need to modify the operating system **View** option to display all files and folders in order to access this path.)
- > **For Linux:** `/var/hasplm`

You can find the **haspvlib\_vendorID.\*** file in the above locations on the computer where Sentinel Vendor Suite is installed.

**NOTE** For information on dependencies between the version of your Vendor library and the version of the Run-time Environment or the Internal /External License Manager, see "Vendor Library Version Dependency" in the [Sentinel LDK Release Notes](#).

## Software for .NET and Java Assemblies

For protected .NET assemblies or Java applications, the following additional files must be distributed with your protected application:

Type of Protected Application	End User Operating System	Additional File Required
.NET assembly	32-bit Windows	haspdnert.dll
	64-bit Windows	haspdnert_x64.dll
Java application	32-bit Windows	HASPJava.dll
	64-bit Windows	HASPJava_x64.dll
	Mac OSX	libHASPJava.dyliblibHASPJava.jnilib
	64-bit Linux	libHASPJava_x86_64.so
	All	Customized Sentinel Licensing API dynamic libraries (copied automatically to the output directory by Sentinel Envelope)

These native library files enable the protected application to communicate with the Sentinel protection key.

**NOTE** For Linux applications that were protected using Sentinel LDK Envelope and that run under Red Hat EL 6.4: The installer for the protected application should determine if libXaw libraries are present on the end user's computer and, if not, install them.

## Network Environment Management

Your end users can manage their network licenses online using Sentinel Admin Control Center, which is part of the Sentinel LDK Run-time Environment. Ensure that you provide them with the URL for accessing this application. For additional information, see ["Sentinel Admin Control Center" on page 229](#).

## Software for Updating Licenses

You can choose to distribute Sentinel Remote Update System to update licenses remotely in deployed Sentinel protection keys. For additional information on this utility, see ["Sentinel Remote Update System \(RUS\)" on page 147](#).

## Firmware for Applications Protected With AppOnChip

For Windows applications protected with AppOnChip, the Sentinel HL key used by AppOnChip must have firmware version 4.52 or later. (This includes .NET assemblies and 64-bit native Windows binaries.) Your customers can upgrade the firmware to the required version by applying the V2C file that you will find on the machine where Sentinel LDK is installed, under:

`%ProgramFiles(x86)%\Thales\Sentinel LDK\Redistribute\Firmware Update\Sentinel HL\Sentinel HL Driverless\`



To update the firmware on the HL (Driverless configuration) key, ensure that the end user receives the V2C file described above. The end user should perform one of the following procedures:

1. Connect the HL key to the machine where the Run-time Environment is installed. No other key should be connected to the machine.
2. In Admin Control Center, use the **Update/Attach** option to apply the V2C file to the HL key.

OR

1. Connect the HL key to the machine where the RUS utility is installed. No other key should be connected to the machine.
2. In the RUS utility, use the **Apply License Update** tab to apply the V2C file to the HL key.

## Distributing Sentinel LDK Run-time Environment

Depending on the type of Sentinel protection key, Sentinel LDK Run-time Environment may be required on the end user's computer to enable your protected software to run by communicating with Sentinel protection keys. (For more information on Sentinel protection keys, see ["Protection Key Attributes" on page 29.](#))

The following sections describe when the Run-time Environment is required and the various options available for distributing the Run-time Environment to your end users.

- > ["Contents of the Run-time Environment" below](#)
- > ["Situations That Require Sentinel LDK Run-time Environment" on the next page](#)
- > ["Installing Run-time Environment With or Without Legacy Drivers" on page 196](#)
- > ["Signing the Run-time Environment Installer" on page 196](#)
- > ["Required Version of the Run-time Environment" on page 198](#)
- > ["Run-time Environment for Windows" on page 203](#)
- > ["Run-time Environment for Mac" on page 206](#)
- > ["Run-time Environment for Linux" on page 207](#)
- > ["Run-time Environment for Android" on page 207](#)

## Contents of the Run-time Environment

The Run-time Environment contains the following elements:

- > Admin License Manager – This is one of several types of license managers available in Sentinel LDK. For details, see ["Types of License Managers" on page 209.](#)
- > Secure storage – An area reserved by Sentinel LDK on the computer's local hard drive. Sentinel SL AdminMode keys are installed in this area. This area can only be accessed or modified by Sentinel LDK components.

- > Admin Control Center – Web-based user interface for Admin License Manager. For details, see ["Sentinel Admin Control Center" on page 229](#).
- > Drivers – Required in certain instances for accessing licenses on HL keys. For details, see ["Installing Run-time Environment With or Without Legacy Drivers" on page 196](#).

## Situations That Require Sentinel LDK Run-time Environment

Sentinel LDK Run-time Environment is a system component that enables communication between a protected application and a Sentinel protection key. Sentinel LDK Run-time Environment also contains Sentinel Admin Control Center, used to manage licenses.

Installation of Sentinel LDK Run-time Environment requires administrator privileges on the target computer. However, on a Windows or Linux computer, Sentinel LDK Run-time Environment is not required for all protected applications.

The tables that follow indicate when a protected application requires Sentinel LDK Run-time Environment to execute.

### Note the following:

- > Sentinel LDK Run-time Environment is always required when any of the following are true:
  - The protected application executes on a Mac machine.
  - The protected application uses the Data File Protection module to encrypt and decrypt data in an external file AND the application was protected using Sentinel LDK v.7.0 or earlier.
  - The application is linked with **libhasp\_windows\_bcc\_vendorld.lib**.
  - In addition, Sentinel LDK Run-time Environment is required on any machine that will access the Sentinel LDK-EMS Customer Portal to perform online activation or online update of protection keys.

## Standalone Licenses

A Standalone license is for a single protected application that executes on the computer where the protection key is located (no concurrency).

The table that follows indicates when a protected application with a Standalone license requires the presence of the Run-time Environment.

Type of protection key	Run-time Environment required on the computer where the protected application executes?
SL AdminMode key	Yes
SL UserMode key	No

Type of protection key	Run-time Environment required on the computer where the protected application executes?
SL Legacy key	Yes
Sentinel HL (Driverless configuration) key	No
Sentinel HL (HASP configuration) key	Yes
HASP HL key	Yes

## Detachable Licenses

To attach a Detached license to a protected application installed on a remote computer, the Run-time Environment must be installed on the remote computer.

## Network Seat Licenses

When a Network Seat license is installed on a given computer, the License Manager on the computer can serve network seats to protected applications than run on the same computer or on remote computers in the same network.

- > The Run-time Environment is required on the computer where the Network Seat license is located.
- > The Run-time Environment is not required on the remote computer where the protected application executes.

The following Sentinel protection keys support Network Seat licenses:

- > SL AdminMode keys with concurrency
- > SL Legacy keys with concurrency
- > HL (HASP configuration) Net and NetTime keys
- > HL (Driverless configuration) keys with concurrency – all types except for Basic

**NOTE** If all of the following are true:

- > A Network Seat license is installed on an HL (Driverless configuration) key.
- > The HL key is located on the same machine as a protected application that consumes a network seat.
- > The local Admin LM is an older version than the Integrated/External LM.

The access attempt may fail with the error `HASP_OLD_LM`.

To ensure full functionality, Thales recommends that you install the latest RTE whenever you update the deployed Integrated/External LM.

## Installing Run-time Environment With or Without Legacy Drivers

On a Windows platform, Sentinel LDK Run-time Environment (RTE) is installed either with or without legacy drivers, depending on the factors described below.

### > Without Legacy Drivers

When the RTE is installed on a new machine, then by default, the installation does not include legacy drivers. This provides the most stable configuration of the RTE. Similarly, when upgrading the RTE on a machine where legacy drivers were not previously installed, the installation does not include legacy drivers.

### > With Legacy Drivers

When the RTE is upgraded on a machine where legacy drivers were previously installed (for example, if RTE 8.15 or earlier was installed), then the installer keeps or upgrades these drivers. This ensures compatibility with existing protected applications.

You have the option of forcing installation of the RTE with legacy drivers on a new machine if necessary. For example, if you know that one of the situations described below exists. This can only be accomplished using Sentinel Run-time Environment Installer API or using the RTE command-line installer (haspdinst.exe). For more information, see the [Sentinel LDK Installation Guide](#).

When forcing installation of the RTE with legacy drivers:

- > On 64-bit systems, **aksdf.sys** is installed.
- > On 32-bit systems, no additional files are installed.

Legacy drivers are required in following situations:

- > When the application was protected using Sentinel LDK Envelope version 6.x or earlier AND the application uses the **Version 1** data file protection mode to encrypt data. In this case, the legacy driver **aksdf.sys** is required.
- > When the application is licensed using a Hardlock parallel port key or using a Hardlock USB key with a very old Hardlock library. In these cases, the legacy driver **hardlock** is required.

**NOTE** The hardlock driver is not supported by RTE 8.41 or later. If your application requires the hardlock driver, you must install RTE 8.31 or earlier.

## Signing the Run-time Environment Installer

Digital signatures allow administrators and end users who are installing the Run-time Environment (RTE) to know that the software is provided by a legitimate publisher. Certain Windows operating systems, for example, enforce the use of digital signatures for some types of code. In these cases, installing unsigned software requires a higher authorization level.

Thales highly recommends that you apply your digital signature to the RTE installer to simplify RTE installation by end users and increase their confidence in your software.

For details on generating a Run-time Environment (RTE) Installer that is customized with your Vendor Codes, see [Sentinel LDK Installation Guide](#).

## Windows

After generating the RTE installer (.exe), you apply your digital signature as follows:

1. Obtain a digital signature from one of the certification authority providers.
2. Prepare a batch (.bat) file that contains the following command:

```
signtool.exe sign /v /p <password> /f <pfxSignatureFile> /n "<subject>" /t  
http://timestamp.verisign.com/scripts/timestamp.dll <rteInstallerExe>
```

where:

- **password**—password for opening the .PFX signature file
- **pfxSignatureFile**—path and name of your signature file
- **subject**—name of the subject of the signing certificate
- **rteInstallerExe**—name of the RTE installer file

For example:

```
signtool.exe sign /v /p pwN#%12A /f abcsoft.pfx /n "ABC Software Inc." /t  
http://timestamp.verisign.com/scripts/timestamp.dll haspdinst.exe
```

3. Place the batch file in the same directory as the **signtool.exe** program and the RTE installer.
4. Run the batch file.

**NOTE** The **signtool.exe** program is provided by Microsoft as part of the Windows SDK.

For more information about driver signing, go to: <https://docs.microsoft.com/en-us/windows/win32/seccrypto/using-signtool-to-sign-a-file>

## Mac

Customized RTE Installers are generated on a non-Mac machine. Therefore, before you distribute the RTE installer to your customers, you must code sign and notarize the RTE Installer for Mac as follows:

1. Code sign the RTE Installer either bundled with your application or separately as a standalone application. If you are code signing the RTE Installer separately, make sure to use:
  - your Apple Developer ID Application certificate.
  - a secure timestamp.
  - "hardened runtime".

- an Apple entitlements file that includes the required connections, such as USB devices and incoming and outgoing network connections. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>com.apple.security.device.usb</key>
    <true/>
    <key>com.apple.security.network.client</key>
    <true/>
    <key>com.apple.security.network.server</key>
    <true/>
</dict>
</plist>
```

2. Notarize the RTE installer by submitting one of the following as a PKG, DMG, or archive file (such as, TAR, XZ, or ZIP) to Apple's notary service:

- The code-signed RTE Installer as a standalone application.
- A bundle that includes both your application and the code-signed RTE Installer.

For more information on code signing and notarization, go to: <https://developer.apple.com/support/code-signing/> and [https://developer.apple.com/documentation/security/notarizing\\_your\\_app\\_before\\_distribution](https://developer.apple.com/documentation/security/notarizing_your_app_before_distribution)

## Required Version of the Run-time Environment

Sentinel LDK uses two mechanisms that determine what version of Sentinel Run-time Environment (RTE) is required on end users' machines (in situations where the RTE is required):

### > Version Enforcement Options (selected in Sentinel Master Wizard)

This mechanism can be used to ensure that customers do not bypass the latest important security and reliability enhancements by installing an earlier version of Sentinel RTE. ("Security" in this case includes the security of the license and of the protected application against vulnerabilities and disassembly.) The version of the RTE on the end user's machines satisfies the minimum requirements for strong security and reliability. The minimum acceptable version of the RTE is set by Thales in each release of Sentinel LDK and is listed in the [Sentinel LDK Release Notes](#).

### > Minimum RTE/API Version (selected in Sentinel LDK-EMS Administration Console)

This mechanism ensures that the version of the RTE on the end users' machines supports important security and reliability features of Sentinel LDK that you want to have included when protecting and licensing your applications. You choose the minimum acceptable version of the RTE based on your requirements.

The installed version of the RTE must satisfy the requirements of both of these mechanisms. For example:

- > If **Minimum RTE/API Version** is 7.60 and **Version-unrestricted** option is selected, minimum version is 7.60
- > If **Minimum RTE/API Version** is 7.60 and **Version-restricted** option is selected, minimum version is 7.90

Each of these mechanisms is described in greater detail the sections that follow.

The following table provides a side-by-side comparison of the mechanisms.

Minimum RTE/API Version	Version Enforcement Options	
	Version-restricted option	Version-unrestricted option
The minimum RTE version is set by the vendor in Sentinel LDK-EMS or in License Generation API.	The minimum RTE version (listed in the <a href="#">Sentinel LDK Release Notes</a> ) is set by Thales in the downloaded API libraries.	There is no minimum version.
Enforcement is set in the License Manager (any type) when a V2C file from Sentinel LDK-EMS or from License Generation API is applied by the License Manager.	Enforcement is set in the application when it is protected by Envelope or by the downloaded API libraries used by the Licensing API	There is no enforcement.
<p><b>Prior to Minimum RTE/API Version 8.11:</b> Version of the License Manager versus the V2C file is checked by the License Manager only when applying the update in the V2C file. Downgrade of the RTE by the end user after applying the license update is not automatically detected.</p> <p><b>Minimum RTE/API Version 8.11 and later:</b> Version of the License Manager versus the V2C file is checked by the License Manager when applying the update in the V2C file and at application runtime. Downgrade of the RTE generates an error.</p>	<p>Version of the License Manager (Run-time Environment) is checked against the version specified in the downloaded Licensing API libraries by the application at run-time.</p> <p>Use of an earlier RTE causes the application generates an error.</p>	There is no enforcement.
Applies when an application has a new or updated license.	Applies when an application is protected or re-protected.	Applies when an application is protected or re-protected.
An error is generated if the application uses a feature that is not supported by the installed RTE.	An error is generated if the application uses a feature that is not supported by the installed RTE.	An error is generated if the application uses a feature that is not supported by the installed RTE.

## Version Enforcement Options

When Sentinel Run-time Environment (referred to as *RTE*) is required for your protected application, it is always preferable to provide the most recent version of the RTE with the new or upgraded application. However, end users require administrator privileges in order to install or upgrade the RTE on their machines. Therefore, you may prefer to allow end users to continue to use an older version of the RTE when they upgrade to a new version of the protected application.

This section describes the options that are available to you for enforcing or disregarding the requirement for a minimum version of the RTE when the RTE is required for the protected application.

API libraries that are customized for your vendor code are used by Sentinel LDK Envelope and Sentinel Licensing API to protect your applications. These libraries are generated by Thales specifically for your Batch Code. You download these customized vendor libraries using the Master Wizard when you introduce one of your Vendor keys.

The Master Wizard offers you a choice of two types of libraries, each incorporating one of the options described below. The option that you select determines how protected applications interact with the Run-time Environment (the RTE). The available options are:

### > **Version-restricted option** (Recommended for best security and reliability in the protected application)

For protected applications that require the RTE: With the version-restricted option, the applications will require a *minimum version* of the RTE (the earliest version that contains the latest important security and reliability enhancements). Use of the version-restricted option ensures that end users cannot downgrade to an earlier version of the RTE and that they use a version of the RTE that provides the best quality together with all the latest security and reliability fixes. This restriction applies both for local deployment of the RTE and for deployment of the RTE on a remote license server machine.

For each new release of Sentinel LDK, the required *minimum version* number is updated only if the RTE for that release contains significant security and reliability enhancements.

For example: The required *minimum version* of the RTE for applications protected with versions 7.9 through 8.0 of the customized vendor libraries remains as RTE version **7.90**, because this version of the RTE contains the latest significant security and reliability enhancements. Later versions of the RTE contain less important enhancements and fixes.

**NOTE** The version-restricted option is only relevant for the static Licensing API because the user can replace the new version of the dynamic Licensing API with an older version.

For example: Given that the dynamic Licensing API version 8.1 has a security issue. The vendor downloads the version-restricted dynamic Licensing API 8.2 using the Master Wizard, and then releases the new version of the dynamic Licensing API. However, a user can bypass any new security enhancements in the new version if they can obtain the old version of the dynamic Licensing API and replace the new version.

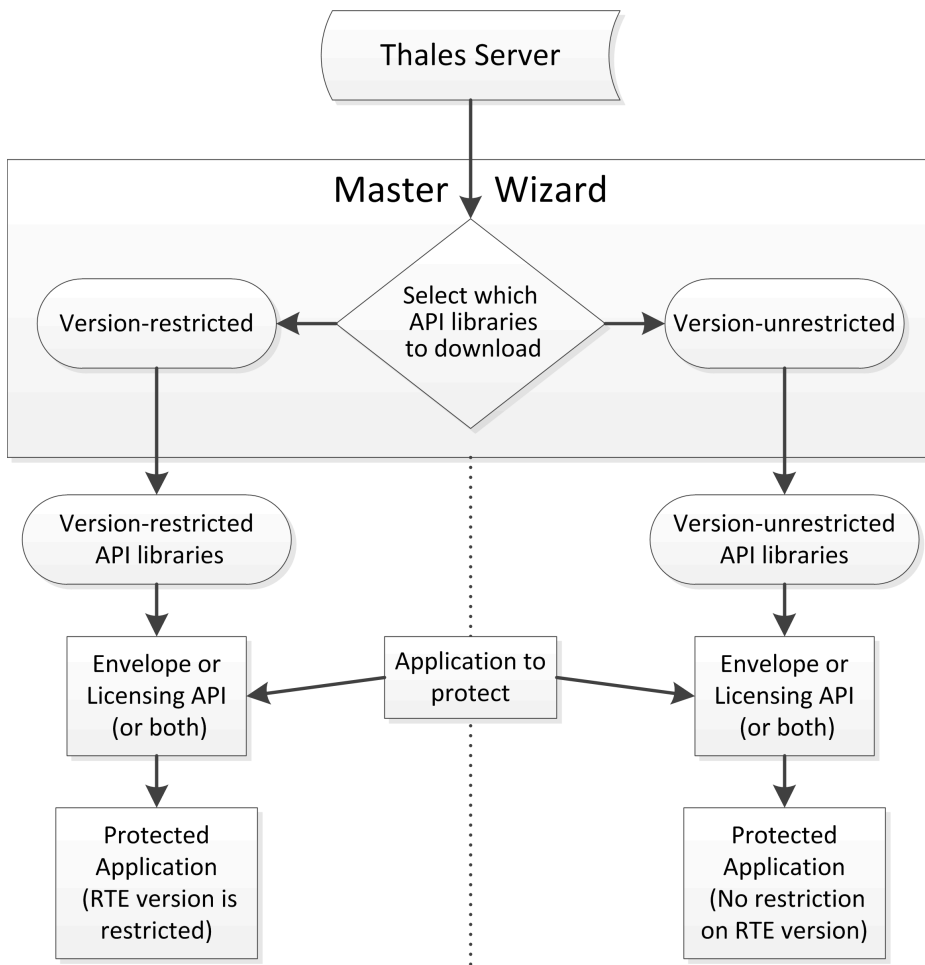


> **Version-unrestricted option** (For compatibility with all versions of the RTE)

For protected applications that require the Run-time Environment: The applications will not check the version number of the RTE. Applications protected with this option can be used with all versions of the RTE. Select this option only if you want to avoid upgrading the RTE at end user sites. This option simplifies deployment, especially when network license servers are used, but does not guarantee that security and reliability fixes in later RTE versions are employed.

**NOTE** With either option, users will need to upgrade their RTE if the protected application uses specific functionalities that require a later version of the RTE.

The diagram that follows illustrates how this process operates.



## Minimum RTE/API Version

The level of security provided by Sentinel LDK for a protected application depends in part on the following components:

- > Version of Sentinel LDK-EMS or of the License Generation API used to generate the license
- > Version of Licensing API libraries used to protect the application (using the Licensing API, Envelope, or both)

- > Version of the Run-time Environment (if any) used to manage and enforce the license (for SL licenses)

(The “security” referred to above consists of the security of the license and the protection application against vulnerabilities and disassembly.)

As a general rule, use of the latest version of each of the components above provides the most advanced and comprehensive security and reliability for the protected application.

However, providing your customers with a protected application and license that incorporates the latest version of all the components is not always practical. For example:

- > Your existing customer base may include applications that were protected and licensed with Sentinel LDK 7.6. You now want to send a license update using Sentinel 8.0. However, the license update includes security features that are not supported by the original 7.6 license. As a result, the updated license may not be valid.
- > When you are ready to issue an updated version of your protected application, you don’t necessarily want to force your customers to update the Run-time Environment at their site. However, you want the application to be protected with the most advanced clone protection schemes that can be supported by the existing RTEs at the customer sites.

You can balance the security and reliability of your applications against the convenience of your customers by choosing the appropriate value for the **Minimum RTE/API Version** configuration parameter in Sentinel LDK-EMS. (For the License Generation API, this value is specified using the `<minimum_rte_api_version>` tag.) The value that you choose determines:

- > the minimum version of the RTE (for SL AdminMode licenses or HL keys with concurrency), and
- > the minimum version of the Licensing API libraries (for SL UserMode licenses)

that must exist on the end users’ machines that will receive the license update.

As a result of the value that you choose:

- > The licenses generated by Sentinel LDK-EMS (or the License Generation API) are compatible with your existing customer base. Only newer security and reliability enhancements that can be supported on the end users’ machines will be implemented.
- > For SL licenses that include the **Platform Default** option, Sentinel LDK will use the most advanced clone protection scheme that is supported by the value that you have selected. (For more information, see ["Using the "Platform Default" Scheme" on page 324.](#))

#### NOTE

- > Regardless of the value you select for this parameter, users will need to upgrade their RTE or Licensing API libraries if the protected application uses specific functionalities that require a later version of the RTE.
- > If you choose the value **8.11** for the Minimum RTE/API Version parameter, RTE version 8.11 *must* exist on the end users' machines.

## Run-time Environment for Windows

The following options are available for distributing Sentinel LDK Run-time Environment for Windows operating systems:

- > Integrate installation of the Sentinel LDK Run-time Environment into your application's installer using either of the two options below:
  - Sentinel LDK Run-time Environment Merge module
  - Sentinel LDK Run-time Environment Installation API
- > Deliver either of the following Sentinel LDK Run-time Environment installation utilities to your end users:
  - `HASPUserSetup.exe`: A GUI-based installer
  - `haspdinst.exe`: A command-line utility

**NOTE** Thales recommends that you not enable end users to mix the use of the `HASPUserSetup.exe` and `haspdinst.exe` utilities. Deliver only one of these utilities to end users.

Each of these methods is described in greater detail in the topics that follow.

## Sentinel LDK Run-time Environment Merge Module

The Sentinel LDK Run-time Environment installation is available as a merge module, in the file `haspds.msm`. You can use the merge module to seamlessly integrate the Sentinel LDK Run-time Environment installation in your MSI installation. Merge modules deliver shared Windows Installer components, code, files, resources, registry entries and setup logic in a single, composite file.

**NOTE** The `haspds.msm` merge module cannot be run as a standalone application.

When integrated with your MSI installer, the `haspds.msm` merge module copies the `haspds_windows.dll` into the Win32 system directory of the end user's computer. The `haspds_windows.dll` is called by the MSI module to install or uninstall the Sentinel LDK Run-time Environment.

The benefits of using the Sentinel LDK installation merge modules in a single unified MSI installer include:

- > Providing end users with a single, compound file for your application that includes the Sentinel LDK Run-time Environment installation
- > Installation self-repair provided by reusing the MSI installer

A demonstration of the use of the `haspds.msm` merge module is provided. For more information, see ["Sample Merge Module Installer" on the next page](#).

## Implementation Requirements

Before including the Sentinel LDK merge module in your installer, review the following requirements:

- > The Sentinel LDK merge module require Windows Installer version 2.0 or later.
- > To successfully execute the Run-time Environment installation, end users require administrator rights. Ensure that this is accounted for in your installation scripts.
- > Processes that require the Sentinel LDK Run-time Environment should not be active in the background when installing the Run-time Environment.
- > Before validating the WSM module, change the project properties to relate to your specific development environment.
- > If you intend to apply a digital signature to your installer, ensure that you first adjust the properties in your development environment.
- > Before compiling the MSI project, change the path to external files to match your development environment

## Implementation of Sentinel LDK Merge Modules

Implementation of Sentinel LDK merge modules is a straightforward process that simply requires you to add the `.msm` file containing the Run-time Environment installation to your MSI-compliant installer setup. After you have created your MSI installer, the wrapped file automatically includes the Sentinel LDK installation merge module.

The `haspds.msm` merge module can be found in:

*%ProgramFiles(x86)%\Thales\Sentinel LDK\API\RuntimeInstall\MSI*

### NOTE

- > Do not alter the versioning data in the default merge module or in the MSI DLL sample.
- > Do not alter any entity in the default merge module.
- > When the Run-time Environment is already installed on a target machine:
  - If you install a version of `haspds_windows.dll` that is newer than the already-installed `haspds_windows.dll`, the installed DLL will be replaced with the new one.
  - If a new version of `haspds_windows.dll` is the same as the previous version, the file timestamp will be compared. If the version of the DLL that is being installed is equal to or older than the existing `haspds_windows.dll`, the DLL will not be replaced.

In any event, the `haspds_windows.dll` will be executed.

## Sample Merge Module Installer

A sample MSI installer containing the Sentinel LDK merge module is provided and should be reviewed before implementing the `haspds.msm` merge module into your own installer.

The sample installer is a full MSI-installer containing the Sentinel LDK Run-time Environment installation merge module and the required shared libraries for installing the Run-time Environment.

The sample installer does the following:

- > Verifies that the user has the required administrator rights to install the Run-time Environment
- > Stops a running Sentinel License Manager service before the Run-time Environment is installed, and re-starts the service after the installation is complete.
- > Installs or removes Run-time Environment

The sample installer can be found in:

*%ProgramFiles(x86)%\Thales\Sentinel LDK\Samples\RuntimeInstall\MSI*

Before attempting to try the sample installer, review the following requirements:

- > Administrator rights are generally required in order to install the Driver sample. However, it is possible for a restricted user to install the Driver. For more information, see Microsoft Support Knowledge Base article # 259459 (<http://support.microsoft.com/kb/259459/en-us>).
- > You must change the resource path to your own environment in the project files (\*.wsi, \*.wsm) in order to successfully compile the samples.

**NOTE** You can incorporate a branded DLL into the sample by replacing the name of the demo DLL with the name of the branded DLL.

## Sentinel LDK Run-time Environment Installer API

Use the Sentinel LDK Run-time Environment installer API to integrate the installation process into your custom setup application. For additional information, see [Sentinel LDK RTE Installer API Reference](#).

### haspdinst.exe

**haspdinst.exe** is a command-line utility that installs the Sentinel LDK Run-time Environment. Following the installation of Sentinel Vendor Suite, the file is located in:

*%ProgramFiles(x86)%\Thales\Sentinel LDK\Redistribute\Runtime Environment\cmd Install*

You can distribute this standalone application to your end users.

#### To install the Sentinel LDK Run-time Environment using haspdinst.exe:

- > At the command-line prompt, type `haspdinst -i`.

For more information and a full list of the available switches for the **haspdinst.exe** utility, see the [Sentinel LDK Installation Guide](#).

## HASPUserSetup.exe

**HASPUserSetup.exe** is a GUI-based installation program to independently install the Sentinel LDK Run-time Environment. Following installation of Sentinel Vendor Suite, the file is located in:

`%ProgramFiles(x86)%\Thales\Sentinel LDK\Redistribute\Runtime Environment\Setup`

This easy-to-use program has an intuitive GUI-based wizard. After your end users run the file, they should follow the on-screen instructions to complete the Run-time Environment installation.

After the successful execution of **HASPUserSetup.exe**, an entry is generated in **Program and Features** in the Windows Control Panel.

## Run-time Environment for Mac

Distribute the Sentinel LDK daemons—**aksusbd** and **hasplmd**—to end users running protected and licensed applications on Mac OS X platforms.

All the Sentinel LDK software for Mac that is required for distribution to end users is provided in the *MacOS/Redistribute/* directory on the Mac machine where you install Sentinel LDK files for Mac

The options for distributing the Mac daemons to end users are described below.

## Installation Using a Multi-Packager

The installation package can be integrated into any multi-package installer that includes the installation for your own application. Include the **Sentinel Runtime Installer.pkg** in the mpkg.

**To locate the Sentinel Runtime Installer.pkg:**

1. In the *MacOS/Redistribute/* directory, double-click **Sentinel Runtime.dmg**. The file opens.
2. Click the **Install Sentinel Runtime Environment** icon and select **Show Original**. The Packages window opens and **Sentinel Runtime Installer.pkg** is displayed.

For additional information, see the **welcome.rtf** file provided in the *MacOS/Redistribute/* directory.

## Installation Using Installer Scripts

Installation scripts are provided in *MacOS/Redistribute/* on the machine where you install Sentinel LDK files for Mac. Open the directory and click **Sentinel Runtime Installer Scripts.dmg**. A new volume named **Sentinel Runtime Installer Scripts** is mounted on your desktop. The volume contains **dinst** and **dunst** files and the *payload/* directory.

You can copy the files in the volume and integrate them in your customized installer. The scripts are not configurable.

For additional information on using the scripts, see the **ReadMe.html** file provided in the **Sentinel Runtime Installer Scripts** volume.

## Run-time Environment for Linux

Distribute the Sentinel LDK daemons—**aksusbd** and **hasplmd**—to end users running protected and licensed applications on Linux Intel and Linux ARM platforms. Without the daemons, the end user's system will not recognize the connected Sentinel protection keys, and the protected applications will not run.

All the Sentinel LDK software for Linux that is required for distribution to end users is provided in the *Linux/Redistribute/* directory on the Linux machine where you install Sentinel LDK files for Linux.

## Using the Installer Scripts to Distribute the Sentinel LDK Daemons

Open the *Linux/Redistribute/Runtime/script* directory. The directory contains **dinst** (install) and **dunst** (uninstall) scripts and the Sentinel LDK Run-time Environment.

You can integrate the scripts in your installer. The scripts are not configurable.

## Using the DEB or RPM File to Distribute the Sentinel LDK Daemons

This option is available for Ubuntu, Debian, SUSE, CentOS, and RedHat Linux Intel.

Open the *Linux/Redistribute/Runtime* directory. The directory contains the Sentinel LDK Run-time Environment and the following files:

- > For 64-bit Intel systems, Ubuntu or Debian: aksusbd\_version\_amd64.deb
- > For 64-bit Intel systems, RedHat, SUSE or CentOS: aksusbd-version.x86\_64.rpm
- > For 32-bit ARM systems, Ubuntu or Debian: aksusbd\_version\_armhf.deb
- > For 32-bit ARM systems, RedHat, SUSE or CentOS: aksusbd-version.armv7hl.rpm
- > For 64-bit ARM systems, Ubuntu or Debian: aksusbd\_version\_arm64.deb
- > For 64-bit ARM systems, RedHat, SUSE or CentOS: aksusbd-version.aarch64.rpm

## Run-time Environment for Android

The protected Android application contains all required distribution files, including the Integrated License Manager. No Run-time Environment is required.

# CHAPTER 18: Sentinel License Manager

Sentinel License Manager is a component of Sentinel LDK that is located on each machine where a protected application executes and on each machine where a protection key is connected. The License Manager enables the protected application to locate and query the protection key that provides licensing authorization for the protected application to operate.

In the most basic configuration, a single License Manager handles the communication between the protected application and a local protection key. In more complex configurations, multiple License Managers may exist on the machine where the protected application exists. These License Managers communicate among themselves and among other License Managers on remote machines where network protection keys are located.

The License Managers provide the protected application with information about the availability of licenses (both local and remote) and manage access to the licenses.

A user or an application can query and configure various aspects of the License Manager functionality. For example, a user can determine where protection keys are located and can control from which machine a protected application consumes a license.

Sentinel Admin API and Sentinel Admin Control Center (a graphical user interface) are available to communicate with the License Managers as described in this section.

**NOTE** For basic configurations, no customization of the License Manager is required at the customer site.

*In this section:*

- > ["Types of License Managers" on the next page](#)
- > ["Selection of the License Manager By the Protected Application" on page 211](#)
- > ["License Manager Tools" on page 214](#)
- > ["Managing User Access to Admin License Manager Information" on page 215](#)
- > ["Managing Access to Standalone and Network Licenses" on page 217](#)
- > ["Returning Network Seats to an SL License" on page 220](#)
- > ["Working Directly With License Manager Configuration Files" on page 221](#)
- > ["Configuring Detachable License Definitions" on page 226](#)
- > ["Making Product Names Visible on the End User's Machine" on page 227](#)
- > ["Loss of Connection With a Network License" on page 228](#)



## Types of License Managers

Several types of License Managers exist, depending on the type of platform used.

### > Integrated License Manager (Windows, Linux Intel, Linux ARM, Android)

The Integrated License Manager (*Integrated LM*) is included in the Sentinel LDK Licensing API and in applications that were protected using Sentinel LDK Envelope. A given instance of the Integrated LM is dedicated to the protected application in which it is included.

The Integrated LM is able to directly handle local SL UserMode keys, local Sentinel HL (Driverless configuration) keys. Admin rights are not required to install the Integrated LM.

The Integrated LM has no user interface. Under Windows, the Integrated LM can be managed with Sentinel Admin API (described in ["License Manager Tools" on page 214](#)). Under Linux, the Integrated LM can be configured manually as described in this section. Under Android, no configuration is required.

The Integrated LM can be upgraded by upgrading the Licensing API or by re-protecting the application with the latest version of Sentinel LDK Envelope.

### > External License Manager (Windows)

The External License Manager (*External LM*) is contained in a standalone file: **hasp\_rt.exe**. The **hasp\_rt.exe** file must be placed in the same directory as the protected application. A given instance of the External LM is dedicated to the protected application whose directory contains the **hasp\_rt.exe** file.

The External LM is able to directly handle local SL UserMode keys, local Sentinel HL (Driverless configuration) keys. (To handle SL UserMode protection keys, you must place your customized Vendor library in the same directory as the protected application.)

The External LM has no user interface. However, the External LM can be managed with Sentinel Admin API.

Admin rights are not required to deploy the External LM. The External LM can be upgraded by simply replacing the **hasp\_rt.exe** file with a later version of the file.

**NOTE** Under certain circumstances, when a protected application fails, the External LM returns network seats to the pool of available seats more quickly than the Integrated LM. For more information, see ["Returning Network Seats to an SL License" on page 220](#).

### > Admin License Manager (Windows, Mac, Linux Intel/ARM)

The Admin License Manager (*Admin LM*) is included as part of the Run-time Environment. The Run-time Environment also includes device drivers, data file encryption drivers, and Sentinel Admin Control Center, which is the user interface for the Admin LM. The Admin LM can also be managed with Sentinel Admin API.

The Admin LM can manage Sentinel HL keys, SL Legacy keys, and SL AdminMode keys. Under Windows, the Admin LM can also manage SL UserMode keys.

Sessions for protection keys that are handled by the Admin LM are visible in Admin Control Center. Sessions for SL UserMode keys are not visible.

The Admin LM must be present on machines where network protection keys are located.

The Sentinel LDK License Manager service must be active at all times on the machine where the Admin LM is used. This service is started automatically when the machine is started.

Installation of the Run-time Environment on a computer requires administrator rights. No special rights are required after the installation.

The table that follows summarizes the differences between the various types of License Managers.

Attribute	Admin License Manager	External License Manager	Integrated License Manager
Supported platforms	Windows, Mac, Linux Intel/ARM	Windows	Windows, Linux Intel/ARM, Android
Management tools	Admin Control Center, Admin API	Admin API	Admin API (Windows only)
Requires admin rights for installation	Yes	No	No
Easily upgradable	Yes	Yes	No
Requires additional files	Yes	Yes	No
Supports Sentinel HL (Driverless configuration) key	Yes	Yes	Yes
Supports Sentinel HL (HASP configuration) key and HASP HL key	Yes	No	No
Supports SL AdminMode key and cloud licensing	Yes	No	No
Supports SL UserMode key	Partial <sup>1</sup>	Yes	Yes
Supports SL Legacy key	Yes	No	No
Supports network key on a license server machine	Yes	No	No

**Legend:**

1 - For more information, see ["Display of Protection Keys and Sessions in Admin Control Center" on page 232](#).

## Selection of the License Manager By the Protected Application

This section describes how a protected application selects a License Manager when more than one type of Sentinel LDK License Manager is available to the protected application.

### Selection of License Manager Under Windows

On a Windows platform, two or more types of License Managers may be available to a protected application. The application selects the License Manager based on the type and location of the protection key that contains the required license. This section describes the process by which the License Manager that will directly access the protection key is selected.

The Integrated LM is always present in a protected application. The External LM is optionally present also. One of the two is always selected to directly access a local protection key or to hand off access requests to a local or remote Admin LM.

The License Managers to directly and indirectly access the protection key are selected as follows:

1. The Integrated LM is selected if the External LM is missing or is an older version. Otherwise, the External LM is selected.
2. A local SL UserMode key is always directly accessed by the selected Integrated/External LM.
3. A local Sentinel HL (Driverless configuration) key is directly accessed by the selected Integrated/External LM if a local Admin LM is absent or is an older version than the Integrated/External LM. Otherwise, access requests are forwarded to the local Admin LM.

**NOTE** If all of the following are true:

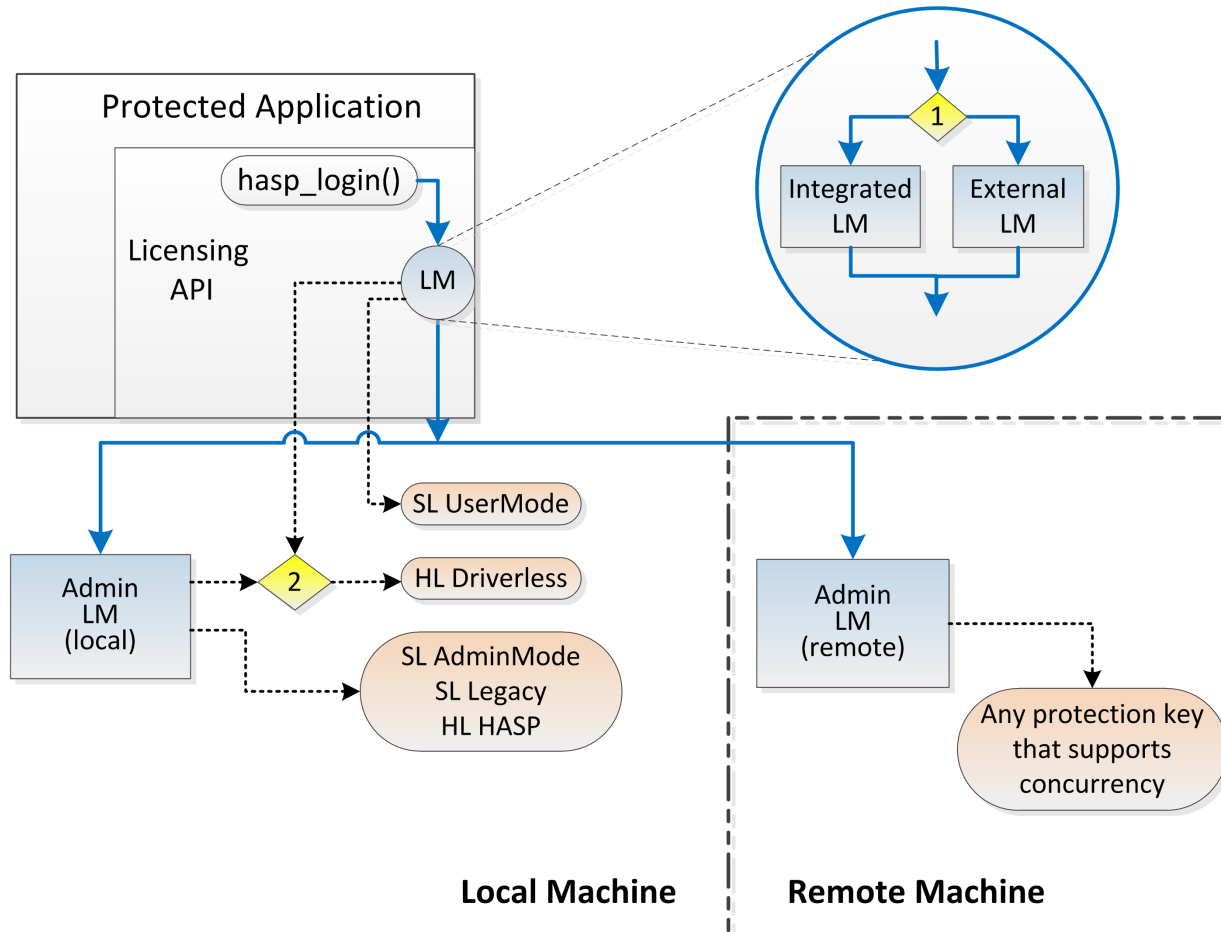
- > A Network Seat license is installed on an HL (Driverless configuration) key.
- > The HL key is located on the same machine as a protected application that consumes a network seat.
- > The local Admin LM is an older version than the Integrated/External LM.

The access attempt may fail with the error `HASP_OLD_LM`.

To ensure full functionality, Thales recommends that you install the latest RTE whenever you update the deployed Integrated/External LM.

4. Access requests for other types of local protection keys are always forwarded to the local Admin LM.
5. When the protection key (of a type that supports network operation) is on a remote machine, the selected Integrated/External LM handles communication directly with the Admin LM on the remote machine, even if a local Admin LM exists. However, if a local Admin LM exists and is active, the Integrated/External LM retrieves the list of license server machines (if such a list exists) from the local Admin LM.

The following diagram shows a graphical representation of the process by which the License Manager to directly access the protection key is selected.



#### Legend:

- 1** The External LM is selected unless the Integrated LM is a more recent version than the External LM or unless the External LM is not present.
- 2** For the Sentinel HL (Driverless configuration) key: The key is directly accessed by the Admin LM unless the selected Integrated/External LM is a more recent version than the Admin LM, or unless the Admin LM is not present.

The following table provides a summary of which License Manager is selected to directly access each type of protection key. Note that the Admin LM and External LM are not necessarily present in all cases.

Type of protection key that contains the required license	Location of the key	License Manager selection priority (from highest to lowest)
Sentinel HL (Driverless configuration) key	Local	<ol style="list-style-type: none"> <li>1. Admin LM (if the Admin LM is the same or a more recent version than the External LM and the Integrated LM)</li> <li>2. External LM (if the External LM is the same or a more recent version than the Integrated LM)</li> <li>3. Integrated LM</li> </ol>
Sentinel HL (HASP configuration) or HASP HL key	Local	Admin LM
SL Legacy key or SL AdminMode key	Local	Admin LM
SL UserMode key	Local	<ol style="list-style-type: none"> <li>1. External LM (if the External LM is the same or a more recent version than the Integrated LM)</li> <li>2. Integrated LM</li> </ol>
A protection key that supports concurrency	Remote	(Remote) Admin LM

## Selection of License Manager for Protected Data Files Under Windows

Selection of the License Manager for data files with licensing protection is determined by the location of the protected application or the Web browser that is used to access the data files, and not by the location of the data files. Therefore, the process for selection of the License Manager is the same as for any other protected application on a Windows platform.

## Selection of License Manager Under Mac

On Mac platforms, only the Admin LM is supported. For a local protection key, all access requests are handled by the local Admin LM. For a remote protection key, the local Admin LM passes the access request to the Admin LM on the remote machine where the protection key is located.

## Selection of License Manager Under Linux

On Linux Intel and ARM platforms, selection of the License Manager is identical to the process for Windows platforms, with one exception: the External LM is not supported and therefore cannot be selected.

## Selection of License Manager Under Android

On Android platforms, only the Integrated LM is supported. The Integrated LM handles all access requests for a local protection key.

The Integrated LM can also consume a network seat from the Admin LM on a remote machine with a protection key that supports concurrency.

## License Manager Tools

Sentinel LDK provides two tools for working with License Managers:

- > Sentinel Admin Control Center
- > Sentinel Admin API

These tools provide the following functionality:

- > Detach a license from a network key and attach the license to your machine or to a different recipient machine
- > Cancel a detachable license prematurely
- > Install an update (V2C file) to a license on a key that is visible to the Admin LM
- > Break down a V2CP file to its component V2C files and then process each V2C file as described above.
- > Generate a C2V file for a Sentinel SL key or a Sentinel HL (Driverless configuration) key that is visible to the Admin LM
- > Configure access permissions from a client machine to a remote license server machine
- > Configuring a license server machine to allow remote access from a client machine
- > Configure detachable licenses parameters and other License Manager parameters

**NOTE** Most of the functionality listed above is relevant only for the Admin LM.

The table that follows indicates the differences between Sentinel Admin Control Center and Sentinel Admin API.

Consideration	Admin Control Center	Admin API
Interface	Web-based graphical user interface	Callable API functions
Target user	Customer's end user or license administrator	Vendor software developer

Consideration	Admin Control Center	Admin API
<b>Types of License Managers handled</b>	Admin LM (Under Windows, Mac, and Linux Intel/ARM)	<ul style="list-style-type: none"> <li>&gt; Under Windows: Admin LM, Integrated LM, External LM</li> <li>&gt; Under Mac: Admin LM</li> <li>&gt; Under Linux Intel/ARM: Admin LM, Integrated LM</li> </ul>

Each of these tools is described briefly below.

## Sentinel Admin Control Center

Sentinel Admin Control Center is a customizable, Web-based, end-user utility that enables centralized administration of Admin LMs and Sentinel protection keys.

Sentinel Admin Control Center is available under the Windows, Mac, and Linux Intel/ARM operating systems.

For more information, see ["Sentinel Admin Control Center" on page 229](#).

## Sentinel Admin API

Sentinel Admin API provides the functionality described above in the form of callable API functions. However, Admin API can be used to access both Admin LMs (under Windows, Mac, and Linux Intel/ARM) and Integrated/External LMs (under Windows only).

You can use Admin API to develop a custom application to replace Admin Control Center.

You can incorporate calls to Admin API in your protected application. This gives you the ability, for example, to make configuration changes in the License Manager each time the protected application executes. Such changes are not dependent on the contents of the License Manager configuration file and, therefore, cannot be modified by the user and are not necessarily visible to the user.

Sentinel Admin API is also accessible in Sentinel LDK ToolBox.

For information regarding Sentinel Admin API, see the [Sentinel Admin API Reference](#).

## Managing User Access to Admin License Manager Information

Admin LMs can be accessed and modified by local and remote users on the network using Sentinel Admin Control Center or (programmatically) Sentinel Admin API.

**NOTE**

- > This topic relates only to accessing information for an Admin LM, such as configuration parameters or Product data. This topic does not relate to accessing local or remote licenses with the Admin LM.
- > No user access parameters exist for Integrated/External LMs. Information for these License Managers is only available to users on the local machine where the License Managers are installed.

By default, only a local end user is authorized to set access rights for the Admin LM on a given machine. Remote users are blocked from accessing any local Admin LM.

## Enabling or Disabling Access for Remote Users

The parameters in the table that follows relates specifically to remote users. If these parameters are set to disable access for remote users, remote users will not be able to access Admin Control Center or Admin API for the local Admin LM. These settings override any access level granted by other parameters described below.

Remote Access Enabled/Disabled	Using Admin Control Center – "Basic Settings" Configuration Page	Using Admin API
Enable access for remote users who are using Admin Control Center	Select the <b>Allow Remote Access to ACC</b> check box.	Set <accremote> to 1.
Disable access for remote users who are using Admin Control Center	Clear the <b>Allow Remote Access to ACC</b> check box.	Set <accremote> to 0.
Enable access for remote users who are using Admin API	Select the <b>Allow Remote Access to Admin API</b> check box.	Set <adminremote> to 1.
Disable access for remote users who are using Admin API	Clear the <b>Allow Remote Access to Admin API</b> check box.	Set <adminremote> to 0.



## Setting the Access Level for Authorized Users

For information in each Admin LM, an authorized end user can set one of the following levels of access for local and remote users who are employing Admin Control Center or Admin API:

Access Level	Using Admin Control Center – "Basic Settings" Configuration Page	Using Admin API
Allow all users to retrieve and modify all information.	For the Password Protection parameter, do not set a password, or change the password to null.	Do not set a password, or use <b>&lt;adminpassnew&gt;</b> to change the password to null.
Allow all users to retrieve all information. Require a password to modify any information.	<ul style="list-style-type: none"> <li>&gt; For the Password Protection parameter, select <b>All ACC Pages</b>.</li> <li>&gt; Set a password.</li> </ul>	<ul style="list-style-type: none"> <li>&gt; Set <b>&lt;passacc&gt;</b> to 1.</li> <li>&gt; Use <b>&lt;adminpassnew&gt;</b> to set a password.</li> </ul>
Allow all users to retrieve all information and to modify all information except for Admin LM configuration parameters. Require a password to modify Admin LM configuration parameters.	<ul style="list-style-type: none"> <li>&gt; For the Password Protection parameter, select <b>Configuration Pages</b>.</li> <li>&gt; Set a password.</li> </ul>	<ul style="list-style-type: none"> <li>&gt; Set <b>&lt;passacc&gt;</b> to 0.</li> <li>&gt; Use <b>&lt;adminpassnew&gt;</b> to set a password.</li> </ul>

**NOTE** The access level described in the table above applies to both local and remote users. However, in practice, the access level is only relevant for remote users because any user that can access the configuration file for an Admin LM can modify access rights manually.

## Managing Access to Standalone and Network Licenses

A network license is Product license that supports concurrency. A network license is typically located on a machine that is different from the machine where the protected application executes.

The following terms are used in this section:

<b>license server machine</b>	A machine on which a network license is located. This machine also contains the Run-time Time Environment. The Sentinel LDK License Manager service must be active on this machine at all times. Multiple license server machines may exist in a network.
<b>client machine</b>	A machine on which a protected application executes.

When a protected application is launched, the License Manager on the client machine initiates a search for a protection key that contains an appropriate Product license. (The client machine may contain more than one License Manager. For more information, see ["Selection of the License Manager By the Protected Application" on page 211.](#))

The customer can configure the way the client machine locates the appropriate license. For example:

- > The customer can configure the License Manager on the client machine or license server machines so that if no license is found locally, the License Manager on the client machine can expand the search to include license server machines in the same network.
- > The customer may want to minimize the time required for a protected application to locate a network license. This can be done by configuring the License Manager on each client machine to access a specific license server machine.
- > An organization may want to set up two or more license server machines in their network and control which end user machines can access each license server machine.

Configuration of the search for a network license can be accomplished using either client-side configuration or license server-side configuration (or any combination of the two).

**NOTE** The scope of the search by a protected application for a Product license on a protection key may be limited by the following additional considerations:

- > The **Protection Key Search Mode** specified for the protected application in Sentinel LDK Envelope may limit the search to the local machine or to remote machines only.
- > The **Locking Type** specified for the protected application in Sentinel LDK Envelope may limit the search to specific types of protection keys.
- > If the Sentinel Licensing API is used in the protected application to log into a specific Feature in a protection key, the login scope may apply a variety of parameters that limit the search for the protection key.

The explanations in this section are based on the assumption that the parameters above are configured to include widest possible scope of protection keys.

## Client-side Configuration

**NOTE** When using client-side configuration, you must also configure the Admin LM on the license server machine to allow remote access from client machines. In Admin Control Center on the license server machine, select the configuration parameter **Allow Access From Remote Clients**.

Use *one* of the following techniques to configure client machines:

- > Use Admin Control Center (if available) on a client machine:

Access the **Configuring Access to Remote License Managers** tab on the Configuration page. Select the **Allow Access to Remote Licenses** option, and use the other parameters on the screen to specify how the client machine should search for remote licenses.

To repeat the configuration on multiple client machines, see ["Working Directly With License Manager Configuration Files" on page 221](#).

> Use Admin API on a client machine:

Use the ContextNewScope() function to establish a context to the Integrated/External LM (or to the Admin LM, if available). Configure which license server machines the client machine should access in order to locate a license for the protected application. Use the <serveraddr> tag for this purpose.

To repeat the configuration on multiple client machines, see ["Working Directly With License Manager Configuration Files" on page 221](#).

> Manually configure the client machine

You can use a text editor to create a Licensing Manager configuration file manually (or copy an existing one). Indicate which license server machines the client machine should access in order to locate a license for the protected application.

For more information, see ["Working Directly With License Manager Configuration Files" on page 221](#).

## License Server-side Configuration

**NOTE** When using license server-side configuration, you must also configure the Admin LM on the client machine to search for remote licenses on license server machines. In Admin Control Center on the client machine, select the configuration parameter **Allow Access to Remote Licenses**.

Use either Admin Control Center or Admin API to configure license server machines.

### To use Admin Control Center:

Use Admin Control Center on each license server machine to do one or both of the following:

- > Access the **Configuring Access From Remote Clients** tab on the Configuration page. Select the **Allow Access from Remote Clients** option. In the **Access Restrictions** field, specify restrictions that limit which remote machines can access the local Sentinel License Manager to consume licenses.
- > Access the **Configuring User Settings** tab on the Configuration page. In the **User Restrictions** field, specify which users, which machines, or which user-machine pairs can access licenses on the license server machine.

To repeat the configuration on multiple license server machines, see ["Working Directly With License Manager Configuration Files" on page 221](#).

**To use Admin API:**

On each license server machine:

1. Use the `ContextNewScope()` function to establish a context to the Admin LM.
2. Configure the Admin LM:
  - a. Use the `<access_restriction>` tag to allow or deny access from specific client machines or subnets.  
*OR*
  - b. Use the `<user_restrictions>` tag to specify which users, which machines, or which user+machine pairs can access the license server machine.

To repeat the configuration on multiple license server machines, see ["Working Directly With License Manager Configuration Files" on the next page](#).

---

## Returning Network Seats to an SL License

---

A protected application that accesses a network seat from an SL license communicates with the Admin License Manager that hosts the SL license using either its Integrated License Manager or an External License Manager (see ["Selection of the License Manager By the Protected Application" on page 211](#)).

When the protected application consumes a network seat from the SL license, the network seat remains allocated as long as the Feature in the application is logged in to the SL key. The network seat is returned automatically to the pool of available seats when one of the following happens:

- > The Feature logs out from the SL key (typically when the protected application exits normally).
- > The login session with the SL key times out due to lack of activity in the protected application. The timeout interval (by default, 12 hours) is defined in the Admin License Manager using Admin Control Center.
- > The login session with the SL key is closed by the License Manager because the protected application terminated abnormally.

After normal termination or timeout, the session is closed immediately.

After abnormal termination:

- > The session is closed immediately if the application uses the External License Manager or there is an Admin License Manager on the same machine.
- > The session is closed after 3 minutes when both of the following are true
  - The application uses the Integrated License Manager (instead of the External License Manager).
  - There is no Admin License Manager on the same machine.

## Working Directly With License Manager Configuration Files

License Manager configuration files can be modified using:

- > Sentinel Admin API (for all types of License Managers)
- > Sentinel Admin Control Center (for Admin API License Managers)

However, under certain circumstance it may be desirable or necessary to work directly with the configuration files using a text editor, as described in this section.

License Manager configuration files do not exist on a given machine until one or more of the following occur:

- > A user submits configuration changes in Admin Control Center.
- > The **writeconfig** command is issued in Admin API to write configuration changes to the configuration file.
- > A configuration file is created manually and placed on the machine.

### Default Location of License Manager Configuration Files

This topic describes the location where each type of License Manager creates or expects to find its configuration file.

For all types of License Managers, this location can be determined in Admin API by retrieving the value of the configuration parameter **<configdir>**.

### Admin License Manager

For the Admin LM on a given machine, the configuration file is called **hasplm.ini**. The pathname of the configuration file is as follows:

- > For Windows x64: *%CommonProgramFiles(x86)%\Aladdin Shared\HASP\hasplm.ini*
- > For Windows x86: *%CommonProgramFiles%\Aladdin Shared\HASP\hasplm.ini*
- > For Linux/Mac: */etc/hasplm/hasplm.ini*

The full path name of the **hasplm.ini** file is displayed in the Diagnostics report in Admin Control Center (see the **INI File** entry).

On a given machine, one **hasplm.ini** file exists for all software vendors who require the Admin LM on the machine.

**NOTE** If you are using Windows in a language other than English, locate the directory in which the common files are stored. (In English Windows, the *Common Files* folder).

## Integrated/External License Manager

For the Integrated LM or External LM, the configuration file is called **hasp\_vendorld.ini**. (*vendorld* is the Vendor ID associated with your Batch Code.) For each account under which a protected application executes on a given machine, the file is placed in one of the following locations:

Type of application	Default Location
Windows Desktop	<i>%LocalAppData%\SafeNet Sentinel\Sentinel LDK\</i>
Service ( <b>Local Service</b> account) x64 operating system	<i>%systemroot%\SysWOW64\config\systemprofile\AppData\Local\SafeNet Sentinel\Sentinel LDK\</i>
Service ( <b>Local Service</b> account) x86 operating system	<i>%systemroot%\System32\config\systemprofile\AppData\Local\SafeNet Sentinel\Sentinel LDK\</i>
Service ( <b>Network Service</b> account)	<i>%systemroot%\ServiceProfiles\NetworkService\AppData\Local\SafeNet Sentinel\Sentinel LDK\</i>
Linux	<i>\$HOME/.hasplm</i>
Android	<p>The INI file should be in the <b>.hasplm/</b> directory in the application data directory as returned by the <code>Android getFilesDir()</code> function.</p> <p>The value returned by this function is typically <i>/data/data/APP_NAME/files/</i> but the value could be different, depending on the Android version and installation.</p>

The Integrated/External LM also searches for configuration information from additional sources, in the following order:

1. (Windows only) The License Manager search for the **hasp\_vendorld.ini** configuration file in the following locations:
  - a. directory where the protected application is installed.
  - b. the **%ProgramData%\Safenet Sentinel\Sentinel LDK\** directory (for applications that were protected with Sentinel LDK 7.6 or later).

This file must be created and maintained manually.

If the **hasp\_vendorld.ini** file is present in more than one of the locations described in this section, the License Manager merges the information in the files. Preference for conflicting information is given to files according to the following priority:

- a. default location
- b. application directory
- c. the **%ProgramData%\Safenet Sentinel\Sentinel LDK\** directory

For example: If files are present in the default location and in the application directory, and both files contain a list of remote license server machines, the License Manager will search first for licenses in the list from the file in the default location. If the two files contain conflicting configuration information, preference is given to information from the file in the default location.

2. If the **Sentinel LDK License Manager** service on the local machine is active and “broadcastsearch” is enabled for the Integrated/External LM, the Integrated/External LM additionally uses the list of remote license server machines from the Admin LM.

**NOTE** An application that is linked with the Borland C static library (that is, *libhasp\_windows\_bcc\_<vendorID>.lib*) does not access the Integrated/External License Manager configuration file. As a result, only default settings are used by the License Manager in this instance.

## Modifying License Manager Configuration Files Manually

You have the option of creating a configuration file manually. This would be typically done when:

- > You want to distribute the same configuration parameters to many machines.
- > You want to place a configuration file in the application directory. A configuration file in this location would be shared by all users who run a protected application on a given machine.

The easiest way to create a configuration file is to copy an existing file that was created using one of the License Manager tools and modify it to suit your requirements.

The configuration file does not have to contain any parameters for which you accept the default values. A typical reason to create a configuration file manually is to specify a remote license server machine. In this case, the file would contain the following entry:

```
serveraddr = remoteServerAddress
```

This parameter is described below in greater detail.

For multiple entries, place each entry on a separate line in the file.

## Additional License Manager Configuration Files Parameters

The table that follows describes configuration parameters that you can insert or modify in the configuration file for any type of License Manager (unless noted otherwise).

Parameter	Description
<code>disable_IPv6</code>	Whether to disable IPv6 protocol. Possible values are: 0 — Do not disable IPv6 protocol. Default. 1 — Disable IPv6 protocol.
<code>serveraddr</code>	Append specific machines that may be searched by the current machine for remote Sentinel License Managers. Specify data as IP addresses (for example: 10.1.1.17), Broadcast addresses (for example: 10.1.1.255), or machine names (for example: hk1m001.ecomp.com). When using the IPv6 protocol, use the IPv6 address format. For example, specify <b>FF02::1</b> to access all remote Sentinel License Managers that are part of the default local group defined in the IPv6 subnet.
<code>requestlog</code>	Whether to generate an access log file. Possible values are: 0 — Do not generate. Default. 1 — Generate.
<code>errorlog</code>	Whether to generate an error log file. Possible values are: 0 — Do not generate. Default. 1 — Generate.
<code>getinfo_uncached</code>	When the GetInfo function or GetSessionInfo function in the Sentinel Licensing API retrieves information about remote keys, the information may be obsolete by several minutes because of caching. (For example, the session/login counters may not be current for several minutes.)  You can disable caching to deliver actual values. However, the additional network request requires significantly more time to retrieve.  Possible values are: 0 — Enable caching. Default. 1 — Disable caching.
<code>load_balancing</code>	Attempt to distribute licensing requests evenly by one of the following: <b>server</b> — Prefer remote License Managers with fewer sessions. Default <b>container</b> — Prefer remote keys with fewer sessions. <b>none</b> — No preference.  Other priorities (license reuse, faster key preference, local key preference) always override these setting.



Parameter	Description
<code>broadcastsearch</code>	<p>Whether to enable the current machine to search for remote Sentinel License Managers on the local network via broadcasts. (Note that the broadcast uses a random UDP source port. This may be an issue with certain firewalls.) If this option is not enabled, every machine on the local network that is to be searched must be specified with the <b>serveraddr</b> parameter (follows below).</p> <p>Possible values are:</p> <p>0 — Disable broadcasts.</p> <p>1 — Enable broadcasts. Default.</p>
<code>emsurl</code>	<p>The URL of a Sentinel LDK-EMS Service that is allowed to contact the Admin License Manager. This URL is only required on machines from which the user will access the Sentinel LDK-EMS Vendor Portal or Sentinel LDK-EMS Customer Portal.</p> <p>The default value is: <code>http://localhost:8080/ems</code></p> <p>However, if Sentinel LDK-EMS Service is located on a remote machine or if it uses a different port number, the URL must be changed accordingly.</p> <p>You can specify multiple <b>emsurl</b> parameters.</p>
<code>port_admin</code>	<p>You can restrict access to Sentinel Admin API and Sentinel Admin Control Center utilizing firewall restrictions. To implement this, specify any port other than 1947 or 80 for the <b>port_admin</b> parameter. For details, see <a href="#">Sentinel Admin API Reference</a>.</p>

For example:

```
disable_IPv6 = 1
requestlog = 0
errorlog = 1
getinfo_uncached = 0
serveraddr = 10.1.1.17
serveraddr = 10.1.1.255
```

## Additional License Manager Files

The table that follows describes additional directories and files that are created by the Admin License Manager under Windows.

The directories and files can be found under the following path:

- > For Windows x64: `%CommonProgramFiles(x86)%\Aladdin Shared\HASP`
- > For Windows x86, under `%CommonProgramFiles%\Aladdin Shared\HASP`
- > For Linux/Mac: `/var/hasplm/`

Directory	Content
<b>attached</b>	The Update/Attach function in Admin Control Center adds a copy of each V2C file that it processes to this directory.
<b>detached</b>	On a license server machine, the License Manager places a copy of each H2R file to this directory when a Detach License action is performed.
<b>rehosted</b>	When an SL key is rehosted from the current machine, a copy of the H2H file generated is placed in this directory.
<b>cancelled</b>	When a detached license is cancelled, a copy of the R2H file is placed in this directory on the recipient machine.
<b>Imid</b>	When Admin License Manager first detects the presence of License Managers on other machines within the same network, it automatically adds an ID file to this directory for each machine detected. The relevant ID file is used by the License Manager when a Detach License action is performed.

The files in the **attached**, **detached**, **rehosted**, and **cancelled** directories are used for logging purposes only. The user can delete these files if necessary.

The ID files in the **Imid** directory can be deleted. However, if the user later want to detach a license to one of the machines, they will have to manually create the ID file for the recipient machine (using Admin Control Center) and place the file in the **Imid** directory on the license server machine.

## Configuring Detachable License Definitions

In Sentinel LDK-EMS, it is possible to flag network-based licenses for Features in Products that will be locked to Sentinel SL keys as being *detachable*. This means that the Product license can be temporarily detached from a pool of network seats and attached to a remote recipient machine for a specific period of time. At the end of the detachment period, the license is automatically restored to the network pool. Prior to the expiration of the license, it is possible to extend its detachment period, or to cancel the detachment and to return the license to the network pool early.

**NOTE** Licenses cannot be detached unless this functionality is enabled as described in this section.

You enable or disable the ability to detach licenses as follows:

- > In Admin Control Center: In the **Detachable License** tab of the Configuration page.
- > In Admin API: Using the <enabledetach> tag.

You can also specify criteria relating to the number of licenses that can be detached from the pool of network seats and the maximum period for which the licenses can be detached. In Admin Control Center, you can specify global settings for all Products, or click the **Per-Product Settings** button to customize settings for individual Products. Global settings will also affect any Products for which individual settings have not been specified.

The Diagnostics page in Admin Control Center on the recipient machine can be used to create a file that containing the machine identity details. This information is required by Admin Control Center on the host machine in order to identify the machine to which a detachable license will be attached.

## Making Product Names Visible on the End User's Machine

When you burn the entitlement for a Product to a Sentinel HL key, the Product name is not necessarily visible in Sentinel Admin Control Center or in Sentinel Admin API on the machine where the Sentinel HL key is connected. The Product name is visible if one of the following actions is performed:

- > You send a V2C file containing an update for the Product. After the user applies the V2C file, the Product name will be visible as long as the Sentinel HL key is connected to the same machine. (If the user moves the key to a different machine, the Product name will not be visible on the new machine.)
- > You export Product names from Sentinel LDK-EMS to an XML file, and place the file on the end user's machine.

**To export Product names from Sentinel LDK-EMS to the end user's machine:**

1. In Sentinel LDK-EMS, click **Developer > Export Catalog Definitions**.
2. In the resulting screen, select the appropriate Batch Code. For **Export File Type**, select **Metadata in Admin Control Center format**.
3. Click **Export**. The file **vendorID.xml** is saved.
4. On the end user's machine, do the following:
  - a. Stop the **Sentinel LDK License Manager** service. (This must be completed before you perform the next step.)
  - b. Place the **vendorID.xml** file in the directory:
 

```
%ProgramFiles(x86)%\Common Files\Aladdin Shared\HASP\vendors\
```

 (For Windows x86, use: %ProgramFiles%\...)
  - c. Restart the **Sentinel LDK License Manager** service.

**To move Product names from one end user's machine to another:**

- > Copy the **vendorID.xml** file from the source machine to the target machine using the procedure in step 4 above.

## Loss of Connection With a Network License

---

A network-type protection key (HL or SL) that contains Features with concurrency typically does not reside on the same machine as the protected application (the client machine).

Under certain circumstances, the communication between the protected application and the protection key may be lost. For example, the protected application may fail or the client machine may crash. As a result, the protection key is left with an open session for a non-existent instance of the protected application. This reduces the number of available network seats for the application in the license.

The Admin LM on the license server machine contains an automatic function that identifies instances where a network protection key and the relevant protected application (on separate machines) have become disconnected. When this situation occurs, the session times out after three minutes. At that point, the Admin LM frees the network seat for re-use.

This functionality is completely automatic and requires no setup or configuration activities by the software vendor or the end user.

# CHAPTER 19: Sentinel Admin Control Center

Sentinel Admin Control Center is a customizable, Web-based, end-user utility that enables centralized administration of Admin License Managers and Sentinel protection keys.

Admin Control Center is designed to provide your end user's system administrator with the means of managing the use of your licensed software by members of the organization. Admin Control Center has been engineered in a way that makes it both flexible and customizable. This makes Admin Control Center a useful add-on to your protected application.

Following are some of the benefits of Admin Control Center:

- > Web-based, meaning that it can be easily accessed from any Web browser. The administrator does not have to be physically present at your end user's site in order to manage the software licenses.
- > Cross-platform capable, enabling it to be used on any platform on which a browser is available.
- > Fully customizable, enabling you to change the displayed information, appearance and behavior so that it will, for example, integrate seamlessly into other applications or match corporate styles. In addition, Admin Control Center can be displayed in a variety of languages.
- > Easy to use, meaning that it can be used with minimal configuration. In addition, the GUI is intuitive, enabling the administrator to manage licenses without the need for a steep learning curve.
- > Enables configuration and control of licenses in a network.

This section describes the functionality, configuration, and customization of Admin Control Center.

All of the functionality that is available in Admin Control Center can also be accessed from any program by calls to the Admin API.

For more information, see ["License Manager Tools" on page 214](#).

*In this section:*

- > ["Launching Admin Control Center" on the next page](#)
- > ["Admin Control Center Interface" on the next page](#)
- > ["Display of Protection Keys and Sessions in Admin Control Center" on page 232](#)
- > ["Administrator's Workflow" on page 233](#)
- > ["Configuration Considerations" on page 233](#)
- > ["Diagnostics" on page 234](#)
- > ["Customizing Admin Control Center Look and Feel" on page 235](#)

# Launching Admin Control Center

Admin Control Center is installed as part of the Sentinel LDK Run-time Environment driver installation process.

Admin Control Center is launched by typing `http://<machine_name or ip_address>:1947` in the address field of the browser. If you are accessing the Sentinel License Manager that resides on your own machine, type: `http://localhost:1947`

**NOTE** Ensure that Sentinel LDK License Manager service is active on the machine where Admin Control Center will run. If Admin Control Center Web pages do not display, see ["Troubleshooting" on page 366](#).

## Admin Control Center Interface

When you launch Sentinel Admin Control Center, the Web interface displays a number of Administration Options on the left of the page. The [Sentinel Admin Control Center help](#) provides information about the fields for each option. Note that the options relate to Sentinel License Manager on the machine whose name or IP address appears in the title bar of Admin Control Center.

Sentinel Admin Control Center

Admin Control Center Help

Sentinel Keys

Products

Features

Sessions

Update/Attach

Access Log

Configuration

Diagnostics

Welcome to the Admin Control Center. This application enables you to manage access to software licenses and Features, to control detachable licenses, to control sessions, and to diagnose problems.

Note: You can select the language in which Admin Control Center is displayed from the bottom of the Options pane.

> The Admin Control Center enables you to monitor the following:

- All the Sentinel protection keys that are currently available on the network server, including their identity, type, and location
- The number of users currently logged in to a protection key, and the maximum number of users allowed to be simultaneously logged into that specific key
- The Features to which each protection key allows access, and any restrictions that apply to the Feature
- The users who are currently logged into a specific protection key, including detailed login information

Note: SL UserMode keys are only displayed for the local (Windows) machine. SL UserMode keys are not displayed when the configuration parameter Do Not Load hasplmrv.exe is selected.

> You can perform actions, such as:

- Detaching a license from the network and attaching it to your machine or a different recipient machine
- Cancelling a detachable license prematurely
- Installing an update to a license on a key that is visible in Admin Control Center

> You can implement and manage cloud licensing.

> You can make basic configuration changes, including:

- Setting the display refresh time
- Configuring access permissions from a client machine to a remote server, and configuring a server to allow it to be remotely accessed
- Defining values for Products with detachable licenses

> The Diagnostics page enables you to view system information related to the current Sentinel License Manager, and to generate reports.

Related Topics

- Cloud Licensing
- Security Considerations
- Sentinel Keys
- Products
- Features
- Sessions
- Update/Attach
- Detach License
- Cancel Detached License
- Access Log
- Configuration
- Diagnostics

Copyright © 2020 Thales Group. All rights reserved.

The following options are available:

- > **Sentinel Keys** enables you to identify which Sentinel protection keys are currently present on the network, including locally connected keys.
- > **Products** enables you to view a list of all the Base Products available on all Sentinel License Managers (local and network). In addition, when a Product contains Features with detachable licenses you can see the number of licenses for the Product that are currently available to be detached from the network and the maximum duration for which they may be detached. This option also enables you to access the Detach/Extend functions.

**NOTE** The Product name for Products that are licensed with Sentinel HL keys are not necessarily displayed in Admin Control Center. For more information, see ["Making Product Names Visible on the End User's Machine" on page 227](#).

- > **Features** enables you to view a list of the Features that are licensed in each of the Sentinel protection keys that are currently present on the network, including locally connected keys. In addition, you can see the conditions of the license, and the current activity related to each Feature.
- > **Sessions** lists all the sessions of clients on the local machine, and those remotely logged in to Sentinel License Manager on the local machine. You can view session data and terminate sessions.
- > **Update/Attach** enables you to update existing licenses on a Sentinel protection key in the field and, in the case of Sentinel SL keys, to attach a detachable license to a recipient machine. It also enables you to apply identification details of an offline recipient machine to a host machine in order to create a file for a detachable license.
- > **Access Log** enables you to view a history of log entries for the server on which Sentinel License Manager is running.
- > **Configuration** enables you to specify certain operating settings for Sentinel Admin Control Center running on the connected machine. You can set parameters relating to user access, access to remote Sentinel License Managers, and access from remote clients. In addition, you can customize log template files in terms of the data they return.
- > **Diagnostics** enables you to view operating information for the Sentinel License Manager to which you are currently logged in, to assist in diagnosing problems. You can generate reports in HTML format. This option also enables you to view miscellaneous data relating to the use of the server on which Sentinel License Manager is running.

For more information about the Diagnostics page, see ["Diagnostics" on page 234](#).

- > **Help** displays the [Sentinel Admin Control Center help](#). Context-sensitive help is available within each of the functions described above, by clicking the **Help** link at the top of the page.
- > **About** provides information about the version of Sentinel License Manager.
- > **Country Flags** enables you to change the language of the user interface. Click the flag of the appropriate country for the language you require.

## Display of Protection Keys and Sessions in Admin Control Center

Information and sessions for protection keys that require the Admin License Manager (*Admin LM*) can be viewed in Admin Control Center.

Typically, protection keys that are handled by the Integrated License Manager (*Integrated LM*) or External License Manager (*External LM*) are located on a machine where the Run-time Environment (and therefore, Admin Control Center) is not installed. As a result, there is no user interface available to display information about the keys or License Manager sessions that use the keys.

However, if Admin Control Center is present on a given machine (even if it is not required):

- > Under Windows, all protection keys that are connected to the machine can be viewed in Admin Control Center.
- > Some of the License Manager sessions for the keys can be viewed in Admin Control Center.

These points are described in more detail below.

### Display of Protection Keys

The table that follows indicates which protection keys are displayed in Admin Control Center when the Run-time Environment is installed on a given machine.

Operating System	Keys That Are Visible in Admin Control Center		
	All HL keys	SL AdminMode Keys	SL UserMode keys
Windows	Yes	Yes	Yes
Linux, Mac	Yes	Yes	No

However, under Windows, certain conditions must be satisfied in order to view local SL UserMode keys in Admin Control Center:

- > The customized Vendor library for the relevant Batch Code must be present on the machine, under:  
`%ProgramFiles(x86)%\Common Files\Aladdin Shared\HASP`  
 To ensure that the Vendor library is present in this location, do one of the following:
  - a. Generate the Run-time Environment installer using Sentinel LDK-EMS.
  - b. Place the Vendor file manually in the location.
- > **hasplmv.exe** must be loaded by the License Manager service. This file is loaded by default. The file is not loaded if the end user specifically enables the option in Admin Control Center to prevent this file from loading.
- > Sentinel Run-time Environment must be version 7.60 or later.



## Display of Sessions

Sessions for all protection keys that are handled by the Admin LM are displayed in Admin Control Center, except for the following:

- > Sessions for local HL (Driverless configuration) keys are displayed in Admin Control only if the Admin LM version is the same or later than the version of the Integrated/External LM.
- > Sessions for SL UserMode keys are not displayed in Admin Control Center.

## Administrator's Workflow

---

When you first launch Admin Control Center, the utility is preconfigured to run automatically. However, you may want to customize it to your requirements and to specify users and their access permissions, and access permissions between remote machines and local servers. Changes to the configuration of Admin Control Center are made in the Configuration tab of the application.

The basic configuration changes that you can make include:

- > Specifying a name for the local machine
- > Enabling access from remote machines to the Admin Control Center web interface on the local machine
- > Setting the display refresh time
- > Defining how many rows of data will be displayed on a page
- > Specifying the logs that are to be created and their content, and customizing information that will be displayed in the log
- > Setting an Admin password

Following the configuration set up, you can define:

- > Users and their access privileges
- > Access parameters to remote Sentinel License Managers
- > Access privileges from remote client machines to a Sentinel License Manager on the current machine

## Configuration Considerations

---

This section briefly describes the main configuration considerations for Admin Control Center.

### Managing User Access to Admin Control Center

You can configure which users can access the Admin Control Center Web interface (or the Admin API) to view or modify information regarding the Admin License Manager. For more information, see ["Managing User Access to Admin License Manager Information" on page 215](#).

## Customizing Log Parameters

You can specify whether Admin Control Center should create an access log and the data that should be included in the log file.

Access the Edit Log Parameters page by clicking **Edit Log Parameters** in the **Basic Settings** tab of the Configuration page.

Additional information about log file parameters is provided in the [Sentinel Admin Control Center help](#).

## Managing Access to Licenses in Admin License Manager

Managing Access to licenses in Sentinel License Manager is performed with the **Users** tab and **Access from Remote Clients** tab in the Configuration page.

For more information, see ["Managing Access to Standalone and Network Licenses" on page 217](#).

## Searching for Sentinel License Managers

The **Access to Remote License Manager** tab in the Configuration page is used determine which locations to include when the local Sentinel License Manager searches for remote Sentinel License Managers.

When you define criteria relating to the machines that may be searched for Sentinel License Manager, you can choose to:

- > Enable a "broadcast" that searches all machines on the local network
- > Search the default local group in an IPv6 subnet
- > Restrict the search to specific machines.

For more information, see ["Managing Access to Standalone and Network Licenses" on page 217](#).

## Diagnostics

---

The Diagnostics page enables you to view and extract operating information for the Sentinel License Manager to which you are currently logged in, to assist in diagnosing problems. You can generate diagnostics reports in HTML format.

Occasionally, it is necessary to create a file containing the machine identity details of a remote recipient machine. This information is required in order for a host machine to identify which machine a detachable license will be attached to. The Diagnostics page enables you to create this file for the local machine on which Admin Control Center is running by using the **Create ID File** button.

Additional information about the data provided in the Diagnostics page is available in the [Sentinel Admin Control Center help](#).

## Customizing Admin Control Center Look and Feel

You can change the language, displayed information, appearance, and behavior of Admin Control Center so that, for example, it will integrate into other applications or match your organization's corporate styles.

The Admin Control Center user interface consists of HTML, GIF, and other files, which are located inside the executable (EXE) file `hasplms.exe`. When you implement additional template sets, you must add them to a fixed directory structure under the **Sentinel LDK** base directory.

**NOTE** As an alternative to customizing Admin Control Center, you can develop your own interface to Admin Control Center functionality by using Sentinel Admin API. For more information, see ["License Manager Tools" on page 214](#).

### To create a directory for a custom template:

1. Locate the `templates` directory inside the *Sentinel LDK base directory*. The location of the Sentinel LDK base directory is as follows:
  - For Windows x64: `%CommonProgramFiles(x86)%\Aladdin Shared\HASPI`
  - For Linux or Mac: `/etc/hasplm/`

**NOTE** If you are using Windows in a language other than English, locate the directory in which the common files are stored. (In English Windows, the *Common Files* folder).

2. Add **<your\_template\_directory\_name>** to the directory. For example, using an English version of Windows 10, the full path is:

`%ProgramFiles(x86)%\Common Files\Aladdin Shared\HASPI\templates\myTemplates`

**NOTE** You can create multiple templates inside your **templates** directory. Each time Sentinel License Manager is launched, the application reads the files in all the directories (except `.bak` files). To expedite the launch time, it is recommended that you keep the directories free of unnecessary files.

3. Restart the Sentinel License Manager.

OR

Call `http://127.0.0.1:1947/action.html?reload_templates` to reload the new template.

To verify your customized template, from a browser on your local machine, open:

`http://127.0.0.1:1947/<yourTemplateDirectory>`.

## Writing Templates

A template is an ASCII text file (typically HTML, but also XML, CSV, or other possibilities) that contains place holders (*tags*) for variables that are inserted by the Admin License Manager when a request is made via HTTP.

In addition, the file may contain block tags that surround a block of text and tags, and generally iterate a list (of Sentinel protection keys, Features, sessions, or other entities).

For example: `{tagname}repeatingblock{/tagname}`

The place holders are written as `{placeholdername}`. For a complete list of available place holder names, their description and usage, see `tagxref.txt` in:

`%ProgramFiles(x86)%\Thales\Sentinel LDK\Docs\Manuals & Tutorials\Admin Control Center Customization`  
(For Windows x86, in: `%ProgramFiles%\...`)

Not all tags work in every context, and some will have different values depending on how they are used. For example, when `{logincount}` is used in a global context, it returns the total login count for the server. When `logincount` is used inside `{devicelist}{/devicelist}`, it returns the login count for the currently-selected Sentinel protection key. If `logincount` is used inside `{featurelist}{/featurelist}`, it returns the login count for the currently-selected Feature.

A special **include** tag is available—`{#include "filename.ext"}`—that will return the contents of a specific file instead of a value. Includes (included files) must not be nested, and must not include a path (meaning that included files must reside in the same directory as the template).

If a table displayed in a browser page returns `*** illegal tag: xxx ***`, the tag is either unrecognized, or is illegal in the current context.

In JavaScript, `{placeholders}` are replaced. To use an opening curly bracket `{`, without it being replaced or generating an *illegal tag* error, ensure that a white space (space, CR, LF, or tab) follows the curly bracket. In this case, it will be passed without modification.

To output something such as `{this}` without it being parsed, use the HTML notation for a curly bracket:  
`&#123;this}`

For additional assistance, refer to the sample templates in the *templates* directory described above.

## Default Templates and Samples

Three sets of template source code are provided:

- > **sample** provides a very simple example of how to use templates and tags.
- > **csv** provides an example for generating a comma-separated (.csv) file for importing to a spreadsheet or database, or for processing by your own program. It produces a CSV list of all available Features.
- > **en** is the complete English-language version of Admin Control Center, as included in the Sentinel License Manager application (`hasplms.exe`). The template uses AJAX technologies to increase ease of use. For translations, or creating a specific corporate identity, use this template set as a starting point.

You can also incorporate some or all of the Sentinel Admin Control Center functionality into your own Web application, possibly with the use of (i)frames or other methods.

## Sample CSV Output

This section provides a sample CSV output. Such output is useful for tasks such as importing the data into spreadsheets or databases.

Using a template such as:

```
c:\>type templates\csv\features.txt
{featurelist}{index}, {hhlid}, {featureid}, "{local}", "{concurrtext}",
{priority},
{fileid}, {filetag}, {logincount}, {loginlimit}, {sessioncount}{/featurelist}
```

The following output is produced:

```
c:\>wget http://10.24.2.23:1947/csv/features.txt -Of.txt & type f.txt
--17:23:44-- http://10.24.2.23:1947/csv/features.txt
=> `f.txt'
Connecting to 10.24.2.23:1947... connected!
HTTP request sent, awaiting response... 200 OK
Length: 1,411 [text/plain]
1, 0x335918F1, 0x00000000, "local", "L", 0, 0xFFCB, 0x0B, 0, 0, 0
2, 0x335918F1, 0x0000BEEF, "local", "LNS", 0, 0x1234, 0x0C, 0, 7, 0
3, 0x335918F1, 0x00001357, "local", "L", 0, 0xABCD, 0x0B, 0, 0, 0
4, 0x335918F1, 0x000CAFF1, "local", "L", 0, 0xCAF1, 0x0B, 0, 0, 0
5, 0x335918F1, 0x000CAFF2, "local", "L", 0, 0xCAF2, 0x0B, 0, 0, 0
6, 0x335918F1, 0x000000A1, "local", "LNS", 0, 0xCAF3, 0x0C, 1, 7, 4
7, 0x335918F1, 0x000000A2, "local", "LNS", 0, 0xCAF4, 0x0C, 0, 7, 0
8, 0x335918F1, 0x0000BEEF, "local", "LNS", 0, 0x1235, 0x0C, 0, 7, 0
9, 0x335918F1, 0x0000BEEF, "local", "LNS", 0, 0x1236, 0x0C, 0, 7, 0
10, 0x335918F1, 0x0000BEEF, "local", "LNS", 0, 0x1237, 0x0C, 0, 7, 0
11, 0x335918F1, 0x0000BEEF, "local", "LNS", 0, 0x1238, 0x0C, 0, 7, 0
12, 0x389C1FAB, 0x00000000, "local", "L", 0, 0xFFCB, 0x0B, 0, 0, 0
13, 0x389C1FAB, 0x00012345, "local", "LNS", 0, 0xAFFE, 0x0C, 0, 7, 0
14, 0x389C1FAB, 0x00055779, "local", "L", 0, 0xBEEF, 0x0B, 0, 0, 0
15, 0x33C90F7A, 0x00011223, "10.24.2.17", "LNS", 0, 0xAFFE, 0x0C, 0, 7, 0
16, 0x33C90F7A, 0x00097531, "10.24.2.17", "LNS", 0, 0x1234, 0x0C, 0, 7, 0
17, 0x33C90F7A, 0x00002FAC, "10.24.2.17", "LNS", 0, 0xCAF2, 0x0C, 0, 7, 0
18, 0x33C90F7A, 0x000AFFEE, "10.24.2.17", "LNS", 0, 0xCAF5, 0x0C, 0, 7, 0
19, 0x33C90F7A, 0x000DFEED, "10.24.2.17", "LNS", 0, 0xCAF9, 0x0C, 0, 7, 0
20, 0x33C90F7A, 0x000FFE01, "10.24.2.17", "LNS", 0, 0x00A1, 0x0C, 0, 7, 0
```

## Configuring Admin Control Center to Use Your Custom Template

After you have created your template, you want to be sure that Admin Control Center loads your customized settings whenever it launches.

By default, when you enter `http://[servername]:1947` in your browser, the internal factory default templates are used. The URL is redirected to `http://[servername]:1947/_int_/index.html`. The characters `_int_` denote the internal directory. If you replace `_int_` with `sample`, the templates from the **sample** directory are used.

To direct Admin Control Center to use your Custom Template:

- 1. Open Admin Control Center in your browser. By default, the application opens at this URL:  
*http://[servername]:1947/\_int\_/index.html*
- 2. In the URL, replace *\_int\_* with the name of the custom template you wish to use.
- 3. Create a shortcut to the address of Admin Control Center with your template.

Using this process, multiple browser windows can use multiple templates simultaneously.

URL Redirections Using HTTP 302

Following is a list of sample URLs to which the browser is redirected when a specific URL is entered.

Note that you do not require this information for translation or simple layout changes in your template. However, it is required if you are changing the logic of Admin Control Center (for example, by adding or removing pages, or merging Admin Control Center functions into another application).

URL Entered	URL Displayed
<b>[server name]:1947</b> Provides a shortcut to the main Admin Control Center page	[server name]:1947/_int_ /index.html
<b>[server name]:1947/corporate.html</b> Automatically switches to the internal template. (_ini_) is set when no template has been specified	[server name]:1947/_int_ /corporate.html
<b>[server name]:1947/csv/devices.txt</b> Does not change because the template (csv) and file name are specified	[server name]:1947/csv/devices.txt
<b>[server name]:1947/sample</b> Automatically redirects to the index.html file when no file name has been specified	[server name]:1947/sample/index.html

**NOTE** It is sufficient to type only the URL of Sentinel Admin Control Center— it automatically redirects to the index page.

## Part 5 - Licensing Business Models

### In this section:

---

- > ["Sentinel LDK Licensing Business Models: Overview" on page 240](#) – Provides an overview of Sentinel LDK Licensing business models.
- > ["Sentinel LDK Licensing Business Models: Description of Models" on page 243](#) – Provides a detailed description of the various Sentinel LDK licensing business models that you can use to distribute your software.

# CHAPTER 20: Sentinel LDK Licensing Business Models: Overview

*In this section:*

- > ["Introduction" below](#)
- > ["Sentinel LDK Licensing" on the next page](#)
- > ["Determining the Best Protection and Licensing Method" on page 242](#)

## Introduction

Today's software industry is more competitive than ever. As with many other industries that once enjoyed exceptionally high margins, software products are increasingly regarded as commodities, with resulting deterioration in both revenues and bottom line profits. To counteract these trends, software publishers and vendors now see the need to change the way they market their products, to increase the value they offer their customers and to better differentiate their offerings from the competition.

Licensing is among the most promising approaches for achieving more-competitive, value-based offerings. Today, software publishers and vendors are seeking ways of moving away from the traditional model—based on perpetual licenses and printed End User License Agreements—toward more flexible licensing business models. New licensing tactics such as trialware, demoware, module- and feature-based licensing, rental, subscription, network licensing—and combinations of these—enable software publishers and vendors to adapt to dynamic markets by offering compelling products that target broader, more segmented markets.

Sentinel LDK is designed specifically to assist software publishers and vendors in pursuit of more competitive product offerings. It not only offers the highest possible level of protection—both against illegal copying and in securing critical intellectual property (IP)—it also enables rapid implementation of novel licensing and distribution models, without the need for extensive engineering of product source code. This enables software publishers and vendors to aggressively extend their market reach and penetration, without negatively impacting their operating margins, to protect the bottom line.

This section describes a wide range of licensing strategies and models designed to provide end users with greater value and additional options for purchasing software products. Using Sentinel LDK's versatile abilities, these strategies and models can be implemented immediately, and can serve as the basis for elaboration and for creating new, tailor-made licensing business models.



## Sentinel LDK Licensing

Sentinel LDK offers a wide range of options and unprecedented flexibility for making and revising both licensing and protection strategies. Virtually any licensing business model can be created—supported by the following fundamental Sentinel LDK concepts, technologies and applications:

> *Protect Once—Deliver Many—Evolve Often™*

The process of protecting software is completely autonomous of marketing and licensing processes, so that after protection has been implemented, diverse licensed products can be created without necessitating changes in the code.

> *Cross-Locking™*

Using Sentinel LDK, the software vendor can choose the device to which the protected software and license are locked—either to one of the many hardware-based Sentinel HL keys, or to a specific computer by means of a versatile software-based Sentinel SL key. The required level of protection, the licensing business model, and the manner in which the software will be accessed and used collectively determine the most appropriate type of Sentinel protection key. Locking the license to a hardware-based Sentinel HL key provides the strongest security.

> Sentinel Remote Update System utility (*RUS utility*)

The *RUS utility* provides a simple and secure method of remotely updating the licenses on deployed Sentinel protection keys. Using the RUS utility, software vendors can renew, extend, revise or revoke a license.

> *LicenseOnChip® and UpdateOnChip*

When a license is supplied on a hardware-locked Sentinel HL key, the licensing logic is embedded in the key's chip, employing Sentinel LDK's patented LicenseOnChip technology. This practice ensures that licenses are hardware-secured and effectively tamper-proof. Likewise, license updates are authenticated in the key's chip.

> *Role-based licensing application*

Sentinel LDK-EMS is a role-based application in which access to each type of task is restricted to authorized personnel. Restricted access provides separation of business activities from order creation, license manufacture and customer follow-up.

> *Versatile Implementation*

Software protection can be implemented using the GUI-driven Sentinel LDK Envelope, the Sentinel Licensing API, or a combination of both. The considerations for choosing a protection method are provided in ["Determining the Best Protection and Licensing Method" on the next page.](#)

> *Detachable Licenses*

A detachable license is available for Products that are locked to Sentinel SL keys in a network environment. Such a license can be temporarily detached from the network pool for use on a remote recipient machine for a defined period.

## Determining the Best Protection and Licensing Method

Sentinel LDK offers two software protection methods that establish an inherent link between the protected software, the license, and the intelligence contained in a specific Sentinel protection key.

> Envelope-based protection (automatic)

Sentinel LDK Envelope automatically wraps software in a protective shield and validates the licensing terms. Sentinel LDK Envelope protection offers ease of use, short time-to-delivery, and anti reverse-engineering features such as file encryption and anti-debugging. It is suitable for protecting compiled executables and DLLs.

> API-based protection (automatic or customized)

Executables or specific functions are protected using Sentinel Licensing API calls that are embedded in the software code. This protection method offers maximum flexibility, and compatibility with a wide variety of development tools and operating systems. API-based protection can be based on predefined Sentinel LDK functions and calls so that licensing terms are validated automatically, or can apply a customized license validation mechanism in order to implement specialized licensing business models.

Most licensing business models discussed in this guide can be applied using either Envelope-based protection or API-based protection. However, some specialized models require customized implementation using the Sentinel Licensing API. Each licensing business model notes the appropriate method or methods.

**NOTE** To enhance the security of your application, when you choose an API-based protection method, it is recommended that you also protect your application with Sentinel LDK Envelope. You can do this using a dedicated Feature ID or with Feature ID 0, which is not linked to a specific license.

For additional information, see ["Preparing Your Sentinel LDK Licensing Plan" on page 98](#).

For information on which important licensing functionality is supported by the various types of protection keys, see ["Protection Key Attributes" on page 29](#). This will assist you in determining which types of protection keys can be used for the various licensing business models described in this section.

## CHAPTER 21: Sentinel LDK Licensing Business Models: Description of Models

This section provides a detailed description of the many types of licensing business models that you can define using Sentinel LDK.

The following table lists the categories and licensing business models available.

Category	Licensing Business Models
"Evaluation Licensing Business Models" on page 245	"Trialware" on page 246 "High-security Time-limited Evaluation" on page 247 "Execution-limited Evaluation" on page 248 "Demoware" on page 250 "License for Testing Protected Applications" on page 251
"Component-based Licensing Business Models" on page 253	"Module-based (Suites)" on page 254 "Feature-based" on page 255
"Metered Licensing Business Models" on page 257	"Time-limited Rental" on page 258 "Phased Rental" on page 259 "Micro-rental" on page 260 "Subscription" on page 261 "Pay-by-Peak Time (Peak Time)" on page 263 "Time-based Overdraft" on page 265 "Standard Counter" on page 266 "Phased Counter" on page 267 "Capacity (CPU/Memory/Disk)" on page 269 "Token-based Licensing" on page 270
"Locked Licensing Business Models" on page 274	"Machine-locked" on page 275 "User-locked" on page 277
"Mobile Licensing Business Models" on page 279	"Portable" on page 280 "Commuter" on page 281 "Software on a Key" on page 282

Category	Licensing Business Models
"Network Licensing Business Models" on page 283	"Limited Concurrent End Users in a Network" on page 284 "Time-limited Concurrent End Users in a Network" on page 286 "Execution-limited Concurrent End Users in a Network" on page 288 "Volume" on page 290 "Site" on page 291
"Sales Boosting Licensing Business Models" on page 293	"KickStart (Quick-delivery Grace)" on page 294 "Referral-based Sales" on page 296 "Automatic Sales Agent" on page 298
"Perpetual Licensing Business Models" on page 300	"Standard Perpetual" on page 301 "Perpetual Unlocked" on page 302

## Evaluation Licensing Business Models

---

Evaluation licensing business models are marketing tools for the software publisher, providing potential end users with the opportunity to test software without making a financial commitment. An evaluation license can be based on fully-functional trialware or on semi-functional demoware. The license can be limited by time or by executions.

When a potential end user subsequently decides to purchase the software, the software vendor can offer any of the paid licensing models described in this guide, with the appropriate key type and locking type. The software vendor uses Sentinel LDK-EMS to create and produce the new license. The evaluation license is then seamlessly converted to a purchased license at the end-user site, using the RUS utility.

The evaluation licensing business models described below are:

- > ["Trialware" on the next page](#)
- > ["High-security Time-limited Evaluation" on page 247](#)
- > ["Execution-limited Evaluation" on page 248](#)
- > ["Demoware" on page 250](#)

## Trialware

<b>Sentinel LDK Functionality</b>	Creates a time-limited, software-based trialware license
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	Sentinel SL (including CL)
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

## Description

Trialware is fully-functional software that is made available for a limited time period (typically between 1 and 90 days) or limited number of executions (typically 30) as a marketing tool. The software is protected with a software-based license, so that it can be distributed both electronically—for example, via a Web site, and on media such as a CD.

The time-limited trialware license does not use a dedicated Sentinel protection key and does not require activation during the trial period. The license is linked to the machine on which the trialware is installed. After the time period expires, the software can no longer run on that machine. However, it can be installed on other machines, creating a super-distribution mechanism when the trialware is referred to others.

## Implementation

1. Select the executable file that you want to license, and determine by which Feature ID it will be identified.
2. Select your protection method:
  - *Envelope-based automatic implementation*  
Protect the executable file using Sentinel LDK Envelope, specifying its Feature ID.
  - *API-based automatic implementation*  
In your code, insert a Sentinel Licensing API Login call to the Feature ID.
3. Create an Unlocked Product in Sentinel LDK-EMS, including the Feature IDs you defined.
4. Distribute your trialware with Sentinel LDK Run-time Environment.
5. When a fully-licensed product is purchased, provide the end user with the appropriate Sentinel protection key programmed with the license.

## High-security Time-limited Evaluation

<b>Sentinel LDK Functionality</b>	Manages the period over which your software can be activated
<b>Software Distribution Method</b>	Physical package
<b>Applicable Key Types</b>	> All Sentinel HL (Driverless configuration) keys except Sentinel HL Basic
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

## Description

The time-limited evaluation software is distributed, protected with a Sentinel HL key for maximum security. Due to the extra cost of providing software with a hardware-based Sentinel HL key, this evaluation method is suitable for high-end software or for software with a high evaluation-to-purchase conversion rate.

## Implementation

1. Select the executable file that you want to license, and determine by which Feature ID it will be identified.
2. Select your protection method:
  - *Envelope-based automatic implementation*  
Protect the executable file using Sentinel LDK Envelope, specifying its Feature ID.
  - *API-based automatic implementation*  
In your code, insert a Sentinel Licensing API Login call to the Feature ID.
3. Create the evaluation Product in Sentinel LDK-EMS and define the expiration date for each Feature ID included in the Product.
4. Distribute the evaluation software with a Sentinel HL key programmed with the license.
5. Create the licensed Product in Sentinel LDK-EMS and define the required licensing terms for each Feature ID included in the Product.
6. When a fully-licensed product is purchased, update the Sentinel HL key using the RUS utility.

## Execution-limited Evaluation

<b>Sentinel LDK Functionality</b>	Manages the maximum number of software executions
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

## Description

Evaluation software that is restricted to a predetermined number of executions. The evaluation software can be distributed with a Sentinel SL key—for example, via a Web site or on a demo CD. Alternatively, it can be distributed with a Sentinel HL key, providing maximum security.

Using a Sentinel HL key for evaluation purposes is usually applicable for high-end software or for software with a high evaluation-to-purchase conversion rate.

When distributing the evaluation software with a Sentinel HL key, the type of key provided must be compatible with the licensing model that will subsequently be applied to the paid license. For example, if the paid license is a rental license, the key used must be a Sentinel HL Time or Sentinel HL NetTime key or must be a Sentinel HL key that supports V-Clock.

## Implementation

1. Select the executable file that you want to license, and determine by which Feature ID it will be identified.
2. Select your protection method:
  - *Envelope-based automatic implementation*  
Protect the executable file using Sentinel LDK Envelope, specifying its Feature ID.
  - *API-based automatic implementation*  
In your code, insert a Sentinel Licensing API Login call to the Feature ID.
3. Create the evaluation Product in Sentinel LDK-EMS and define the permitted number of executions for each Feature ID included in the Product.
4. Distribute the evaluation software with a Sentinel protection key programmed with the license.
5. Create the licensed Product in Sentinel LDK-EMS, defining the licensing terms for each Feature ID included in the Product.



6. When the end user purchases a fully-licensed product, update the Sentinel protection key using the RUS utility.

## Demoware

<b>Sentinel LDK Functionality</b>	Manages active and inactive software functionality
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	API-based automatic implementation

## Description

The demo version of the software is limited to a subset of the functions provided in the fully-licensed product. Demoware can be distributed either with a Sentinel SL key (for example via a Web site or on a demo CD), or with the superior protection of a Sentinel HL key.

Demoware provides prospective end users with limited software functionality, at no charge. Even if the end user does not subsequently purchase the software, the demoware is not discarded, serving as a constant reminder that more powerful functionality can be purchased, with your brand name at the forefront.

**NOTE** When distributing the demoware with a Sentinel HL key, the type of key provided must be compatible with the licensing model that will subsequently be applied to the paid license. For example, if the paid license is a rental license, a Sentinel HL Time or Sentinel HL NetTime key must be used or the key must support V-Clock.

## Implementation

1. Select the software functions that you want to license separately, and determine by which Feature ID they will be identified.
2. In your code, insert a Sentinel Licensing API Login call to the Feature ID.
3. Create two Products in Sentinel LDK-EMS:
  - The demoware Product, including only those Feature IDs that are designated for the demoware. Define a Permanent license for these Features.
  - The fully-licensed Product, including the full set of Feature IDs. Define the required license terms for these Features.
4. Envelope your software for additional security (optional).
5. Distribute the demoware.
6. When the end user purchases the software, send a Sentinel protection key programmed with the full license.

## License for Testing Protected Applications

<b>Sentinel LDK Functionality</b>	Issues cloud-based licenses that enable internal users to test protected applications without the risk that unauthorized users will be able to run the applications.
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	Sentinel CL
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

### Description

To enable testing of a protected application, software vendors typically hosts a network license in their LAN and allows internal users such as QA to use it for testing and education.

If a vendor wants to prevent unauthorized users from accessing the license, they can use a cloud-enabled SL license to grant access to the protected application, and provide only authorized users with the required client identities to run the application. The vendor can further limit the validity of the client identities by including an expiration date and restricting the client identity to the relevant application.

### Implementation

1. Determine which executable file you want to license, and determine by which Feature ID it will be identified.
2. Select your protection method:
  - *Envelope-based automatic implementation*  
Protect the executable file using Sentinel LDK Envelope, specifying its Feature ID.
  - *API-based automatic implementation*  
In your code, insert a Sentinel Licensing API Login call to the Feature ID.
3. In Sentinel LDK-EMS:
  - a. Create a Product that includes the Feature ID. Set the license type to **Time Period** or **Expiration Date**. Include:
    - Parameters for the license expiration.
    - Concurrency counter to the required maximum number of concurrent licenses, and determine whether concurrent instances will be counted for each station, each login or each process.

**TIP** You can specify the licensing parameters each time an entitlement is created. This enables you to use the same Product to produce licenses for testing different applications.

- b.** Use **Produce & Push** to install the SL license on your Cloud license server.
- 4.** Use Sentinel Cloud Portal to generate a client identity for the Product.
- 5.** Distribute the client identity and the protected application to the relevant personnel in your organization.

## Component-based Licensing Business Models

---

Often, software vendors do not want to sell all the software functionality as a single package, preferring to mix and match components in order to create different offerings. Using Sentinel LDK, software vendors have complete freedom to determine the granularity of licensed items, at the level of a specific functionality or component, or at the level of an executable file.

The component-based models described below are:

- > ["Module-based \(Suites\)" on the next page](#)
- > ["Feature-based" on page 255](#)

## Module-based (Suites)

<b>Sentinel LDK Functionality</b>	Manages licensing of individual executables
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

## Description

Each module (executable file) is licensed separately. Assorted software can be bundled into a suite, including software from other software vendors. The license for the entire suite is supplied on a single Sentinel protection key.

## Implementation

1. Select the executable files that you want to license separately, and determine by which Feature ID they will be identified.
2. Select your protection method:
  - *Envelope-based automatic implementation*  
Protect the executable file using Sentinel LDK Envelope, specifying its Feature ID.
  - *API-based automatic implementation*  
In your code, insert a Sentinel Licensing API Login call to the Feature ID.
3. In Sentinel LDK-EMS:
  - a. Create one or more Products.
  - b. Include the required Feature IDs in each Product.
  - c. Define the appropriate license terms for each Feature—for example, the number of executions, expiration date or concurrency.
4. Distribute your software suite with the appropriate Sentinel protection key programmed with the license.

## Feature-based

<b>Sentinel LDK Functionality</b>	Manages licensing of separate functional components
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	API-based automatic implementation

## Description

Software components or functionality are licensed separately, without necessitating changes in the code. Feature-based licensing can be useful in many different scenarios.

### > *Example 1: Basic Software with Add-ons*

Your basic software is provided with a perpetual license. Additional features are licensed separately, and are available at a charge.

### > *Example 2: Software Levels*

Different levels of your software are offered—for example, Student, Light, Standard, and Professional versions. The protection method determines which components are active in each version.

### > *Example 3: Customized Software*

Your software is customized to display or hide functionality depending on the requirements of different end users.

### > *Example 4: Skins or Themes*

The end user is able to choose from a selection of skins or themes, or a user-specific design is created and applied.

## Implementation

1. Select the software functions that you want to license separately, and determine by which Feature ID they will be identified.
2. In your code, insert a Sentinel Licensing API Login call to each Feature ID.
3. In Sentinel LDK-EMS:
  - a. Create one or more Products.
  - b. Include the required Feature IDs in each Product.
  - c. Define the appropriate license terms for each Feature—for example, number of executions, expiration date or concurrency.

4. Envelope your software for additional security (optional).
5. Distribute your software with the appropriate Sentinel protection key programmed with the license.



## Metered Licensing Business Models

---

In recent years, licensing business models that are based on usage, rather than providing an end user with ownership of the software, have become more prevalent. These models all apply some form of metering, the most common of which are rental (time-based) and execution (counter-based) metering. Some models require a prepaid fee, while others enable payment for each use. The models in this section include:

- > *Rental packages* —Time-limited rental, phased rental, micro-rental, subscription.

In this group of models, the license is prepaid or paid on a monthly basis. When it expires, the end user can only continue using your software by extending the license.

- > *Pre-paid execution-based packages*— Standard counter and phased counter.

The license provides a prepaid number of executions. When these have been consumed, the end user must purchase a new package of executions.

- > *Specialized packages*— Capacity, pay-by-peak time, time-based overdraft, execution-based overdraft, token-based licensing.

The metered models described below are:

- > ["Time-limited Rental" on the next page](#)
- > ["Phased Rental" on page 259](#)
- > ["Micro-rental" on page 260](#)
- > ["Subscription" on page 261](#)
- > ["Pay-by-Peak Time \(Peak Time\)" on page 263](#)
- > ["Time-based Overdraft" on page 265](#)
- > ["Standard Counter" on page 266](#)
- > ["Phased Counter" on page 267](#)
- > ["Capacity \(CPU/Memory/Disk\)" on page 269](#)
- > ["Token-based Licensing" on page 270](#)

## Time-limited Rental

<b>Sentinel LDK Functionality</b>	Manages the time period over which your software can be used
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL (Driverless configuration) keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

## Description

The end user pre-pays a fee for a specific period of time, either for a predetermined number of days or terminating on a predetermined expiration date.

End users can monitor the remaining time using Sentinel Admin Control Center, and can order a license renewal before the license expires. License renewal is implemented using the RUS utility.

**NOTE** You can also specify a licensing period that is shorter than one day, as described in ["Micro-rental" on page 260](#).

## Implementation

1. Select the executable file that you want to license, and determine by which Feature ID it will be identified.
2. Select your protection method:
  - *Envelope-based automatic implementation*  
Protect the executable file using Sentinel LDK Envelope, specifying its Feature ID.
  - *API-based automatic implementation*  
In your code, insert a Sentinel Licensing API Login call to the Feature ID.
3. In Sentinel LDK-EMS, create a Product that includes the Feature ID and define either an expiration date or the number of days until expiration.
4. Distribute your software with the appropriate Sentinel protection key programmed with the license.
5. Renew the license remotely using the RUS utility.

## Phased Rental

<b>Sentinel LDK Functionality</b>	Manages the time period over which your software can be used
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL (Driverless configuration) keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

## Description

The end user pays a monthly fee, with a phased pricing structure, which can be associated with an entire product or a specific functionality. The transition from one phase to another is implemented using the RUS utility.

- > **Phase 1** : A fraction of the regular usage price is charged (micro-payment) for a limited period of time. This provides an incentive for the end user to enter into a rental agreement for use of the software. If payment is not received for Phase 2, the license expires at the end of the defined time period.
- > **Phase 2** : The full monthly rental price is charged, for an indefinite time period.

## Implementation

1. Select the executable file or software functions that you want to license, and determine by which Feature ID each file or function will be identified.
2. Select your protection method:
  - *Envelope-based automatic implementation*  
Protect the executable file using Sentinel LDK Envelope, specifying its Feature ID.
  - *API-based automatic implementation*  
In your code, insert a Sentinel Licensing API Login call to the Feature ID.

**NOTE** To set the time limit for a specific functionality, apply API-based automatic implementation. To set the time limit for an executable file, apply either Sentinel LDK Envelope-based or Sentinel Licensing API-based automatic implementation.

3. In Sentinel LDK-EMS, create a Product that includes the Feature ID and define an expiration date or the number of days until expiration of Phase 1.
4. Distribute your software with the appropriate Sentinel protection key programmed with the license.
5. Subject to receiving payment for Phase 2 from the user, extend the license remotely using the RUS utility.

## Micro-rental

<b>Sentinel LDK Functionality</b>	Manages the time period over which your software can be used
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL (Driverless configuration) keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	API-based automatic implementation

## Description

The end user purchases a predefined number of “usage hours.” When the hours are consumed, a new package of hours is purchased.

## Implementation

1. Select the executable file that you want to license, and determine by which Feature ID it will be identified.
2. In your code, insert a Sentinel Licensing API Login call to the Feature ID.
3. Determine what constitutes “active” for the purpose of counting usage and define this in your code, for example:
  - Your software window is focused and activity is detected.
  - Your software is active, performing calculations, even if the window is not focused.
4. In Sentinel LDK-EMS, in the Protection Key memory, define the total number of software activity hours that has been purchased.
5. Envelope your software for additional security (optional).
6. Distribute your software with the appropriate Sentinel protection key programmed with the license.
7. Using the Sentinel Licensing API and the key’s built-in clock:
  - a. Calculate the accumulated active time.
  - b. Write the result to the Protection Key memory.
  - c. Verify that the accumulated time has not exceeded the number of purchased hours.
  - d. When the number of purchased hours is about to expire, display a warning message.
8. When payment is received for additional usage, renew the license remotely using the RUS utility.

## Subscription

<b>Sentinel LDK Functionality</b>	Creates an unconditional license that can be updated remotely
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL (Driverless configuration) keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

## Description

The end user pays a monthly subscription fee that covers the initial software package plus periodical updates. If the end user does not renew the subscription, the basic package and all paid updates remain the property of the end user. New updates are not provided.

## Implementation

1. Select the executable file that you want to license, and determine by which Feature ID it will be identified.
2. Select the protection method for your software:
  - *Envelope-based automatic implementation*  
Protect the executable file using Sentinel LDK Envelope, specifying its Feature ID.
  - *API-based automatic implementation*  
In your code, insert a Sentinel Licensing API Login call to the Feature ID.
3. In Sentinel LDK-EMS, create a Product that includes the Feature ID for your initial software and define a perpetual license for the Feature.
4. Create a component in your software that manages the installation of software updates, and assign it a Feature ID. Select and implement your protection method for that component (Sentinel LDK Envelope or Sentinel Licensing API-based).
5. In Sentinel LDK-EMS, create a Product that includes the Feature ID for the update-installation component and define an expiration date for that Feature.
6. Envelope your software for additional security (optional).
7. Distribute your software with the appropriate Sentinel protection key programmed with the license.

8. During the subscription period, use the RUS utility to send updates to the subscriber. The updates are handled by the update-installation component in your software. Optionally, use Sentinel LDK to encrypt the update files so that the Sentinel protection key is required to decrypt them.
9. Continue sending updates as long as the end user's subscription is valid.
10. When the end user renews the subscription, use the RUS utility to update the expiration date for the update-installation component's license.

## Pay-by-Peak Time (Peak Time)

<b>Sentinel LDK Functionality</b>	Compares a value in the Protection Key memory with a value collected during run-time
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL (Driverless configuration) keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	API-based automatic implementation

## Description

The end user purchases a predefined number of “usage units”. Differential charging is calculated according to the hour of the day or the day of the week in which your software is used. When your software is used at peak demand time, more “usage units” are consumed than at low demand time. This type of license might be applicable in an environment such as a learning facility, in order to encourage students to use resources at low demand time.

## Implementation

1. Select the executable file that you want to license, and determine by which Feature ID it will be identified.
2. In your code, insert a Sentinel Licensing API Login call to the Feature ID.
3. Determine what constitutes “active” for the purpose of calculating usage and define this in your code, for example:
  - a. Your software window is focused and activity is detected.
  - b. Your software is active, performing calculations, even if the window is not focused.
4. In Sentinel LDK-EMS, in the Protection Key memory, define the total number of “usage units” that has been purchased and the pricing structure (number of “usage units” for each time unit and each rate).
5. Envelope your software for additional security (optional).
6. Distribute your software with the appropriate Sentinel protection key programmed with the license.
7. Using the Sentinel Licensing API and the key’s built-in clock:
  - a. Calculate the accumulated active time for each separate rate.
  - b. Calculate the total number of “usage units” consumed.
  - c. Write the result to the Protection Key memory.
  - d. Verify that the accumulated consumption has not exceeded the total number of “usage units” defined in the Protection Key memory.

- e. When the “usage units” are about to expire, display a warning message.
- 8. Using the RUS utility, replenish the pool of “usage units” when the license is renewed.



## Time-based Overdraft

<b>Sentinel LDK Functionality</b>	Manages the time period over which software can be used
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL (Driverless configuration) keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	API-based automatic implementation

## Description

A differential pricing structure is implemented, in which a nominal price is charged for use of your software until a defined expiration date. Following expiration, a higher price may be charged for a limited period, to enable the end user to continue using your software until the license is renewed.

## Implementation

1. Select the executable file that you want to license, and determine by which Feature ID it will be identified.
2. In your code, insert a Sentinel Licensing API Login call to the Feature ID.
3. In Sentinel LDK-EMS, create a Product that includes the Feature ID and define either an expiration date or the number of days until expiration. Include both the regular usage period and the overdraft period in the time that you define.
4. Envelope your software for additional security (optional).
5. Distribute your software with the appropriate Sentinel protection key programmed with the license.
6. Using the Sentinel Licensing API and the key's built-in clock:
  - a. Calculate the time period.
  - b. When the regular usage period terminates, display a message informing the end user that the usage is now subject to overdraft terms and state the expiration date of the overdraft period.
  - c. When the end user renews the license, billing includes payment for the overdraft usage in addition to the license renewal.
  - d. After payment has been received, renew the license remotely using the RUS utility.

## Standard Counter

<b>Sentinel LDK Functionality</b>	Manages the maximum number of software executions
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

## Description

The end user purchases a predefined number of software executions, which can be defined for your software or for specific functionality. An execution-based license might appeal to end users who use your software or a software functionality sporadically, and prefer to pay only when they actually run your software or use the functionality.

End users can monitor the remaining executions using Sentinel Admin Control Center, and can order a license renewal before the license expires. The license renewal is implemented using the RUS utility.

## Implementation

1. Select the executable file or software function that you want to license, and determine by which Feature ID the file or function will be identified.
2. Select your protection method:
  - *Envelope-based automatic implementation*  
Protect the executable file using Sentinel LDK Envelope, specifying its Feature ID.
  - *API-based automatic implementation*  
In your code, insert a Sentinel Licensing API Login call to the Feature ID.

**NOTE** To set a counter for a specific functionality, apply API-based automatic implementation. To set a counter for an executable file, apply either Sentinel LDK Envelope-based or Sentinel Licensing API-based automatic implementation.

3. In Sentinel LDK-EMS, create a Product that includes the Feature ID and define the number of executions.
4. Distribute your software with the appropriate Sentinel protection key programmed with the license.
5. Renew the license remotely using the RUS utility.

## Phased Counter

<b>Sentinel LDK Functionality</b>	Manages the maximum number of software executions
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

## Description

The end user purchases a predefined number of software executions, which can be associated with all of your software or a specific functionality. The pricing structure is phased, and the transition from one phase to another is implemented using the RUS utility.

- > **Phase 1:** For a limited number of executions, the end user pays a fraction of the regular usage price (micro-payment). This provides an incentive for the end user to start purchasing executions. If payment is not received for Phase 2, the license expires when these executions have been consumed.
- > **Phase 2:** The end user pays the regular price for each software execution.

## Implementation

1. Select the executable file or software function that you want to license, and determine by which Feature ID the file or function will be identified.
  2. Select your protection method:
    - *Envelope-based automatic implementation*  
Protect the executable file using Sentinel LDK Envelope, specifying its Feature ID.
    - *API-based automatic implementation*  
In your code, insert a Sentinel Licensing API Login call to the Feature ID.
- NOTE** To set a counter for a specific functionality, apply API-based automatic implementation. To set a counter for an executable file, apply either Sentinel LDK Envelope-based or Sentinel Licensing API-based automatic implementation.
3. In Sentinel LDK-EMS, create a Product that includes the Feature ID and define the number of executions included in Phase 1.
  4. Distribute your software with the appropriate Sentinel protection key programmed with the license.

5. Subject to receiving payment for Phase 2 from the end user, replenish the number of executions remotely using the RUS utility.

## Capacity (CPU/Memory/Disk)

<b>Sentinel LDK Functionality</b>	Manages resource usage
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	API-based automatic implementation

## Description

License consumption depends on utilization of resources—for example, CPU usage or disk space. The more resources the end user consumes, the sooner the license runs out. This type of license might be applicable in an environment such as a learning facility, in order to limit the resources consumed by students.

## Implementation

1. Select the executable file that you want to license, and determine by which Feature ID it will be identified.
2. In your code, insert a Sentinel Licensing API Login call to the Feature ID.
3. Determine the parameters for calculating software usage, and define them in your code, for example:
  - CPU activity related to your software.
  - Disk space usage each time a file is saved from your software.
4. In Sentinel LDK-EMS, create a Product that includes the Feature ID and define the license terms—for instance, a perpetual license or a time-limited license.
5. In Sentinel LDK-EMS, in the Protection Key memory, define the capacity that has been purchased.
6. Envelope your software for additional security (optional).
7. Distribute your software with the appropriate Sentinel protection key programmed with the license.
8. Using the Sentinel Licensing API:
  - a. Calculate the accumulated usage.
  - b. Write the result to the Protection Key memory.
  - c. Verify that the accumulated usage has not exceeded the purchased capacity.
  - d. When purchased capacity has almost expired, display a warning message.
9. When payment is received for additional usage, renew the license remotely using the RUS utility.

## Token-based Licensing

<b>Sentinel LDK Functionality</b>	Provides access to a portfolio of applications or functions using a single license. Charge a different price for each application or function based on the value of the application or function.
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL (Driverless configuration) keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	API-based implementation

### What is Token-based Licensing?

Token-based licensing is a flexible, metered licensing model in which you license a portfolio of applications by providing a pool of "tokens" to the customer.

The tokens provide access for users in the customer organization to a number of applications in your portfolio. Tokens are similar to executions for the **Execution Count** license type. However, each application in the portfolio can be configured to consume a different number of tokens from the pool, depending on the value that you assign to the application.

You can also use token-based licensing to charge for the use of specific functions in your applications.

In either case, the customer organization requires only a single pool of tokens. The customer must purchase a number of tokens based on the applications that they expect to use and the number of end users that will work with each application.

### What Are the Advantages of Token-based Licensing?

Token-based licensing provides advantages for both the customer and the vendor.

#### Advantages for the Customer

- > If the customer requires different applications from the portfolio at different times or at different phases of a project, they can use their pool of tokens to access any of the various applications in the portfolio with no need to purchase different licenses.
- > Token-based licensing significantly reduces the frequency of license renewals that are commonly required when dealing with a large number of products.

#### Advantages for the Vendor

- > The use of tokens reduces sales friction. They eliminate bottlenecks, simplify order transactions, and enable you to conclude sales quickly.

- > You can modify the number of tokens consumed by any application at any time. Thus, for example, you can offer discounted prices for a limited time for a special promotion or for the introduction of a new product.

## Implementation

**NOTE** A number of different methods can be used with Sentinel LDK to implement token-based licensing. This section describes one possible method.

This section describes the steps to use to set up and work with token-based licensing.

### 1. Create a Price List and Token Pool for the Available Applications

Define a Product (referred to as the “**Tokens** Product”) as described below. This Product license will reside on a license server in each customer's network. The **Tokens** Product is used to:

- Store the price for each application or function that you want to include in your portfolio for the customer. The price list indicates how many tokens the customer is charged for the execution of each application or function.
- Store the central token pool for the customer. The tokens purchased by the customer are added to this pool. Each time that a customer executes an application or function from the portfolio, the appropriate number of tokens are consumed from this pool.

#### To create a price list for the available applications:

- Define a Feature in the Sentinel LDK-EMS catalog for each application or function that you want to license using the token-based licensing model.
- Define the **Tokens** Product (described above) to contain the price list and the central token pool for all the available applications and functions.
- Add each of the Features defined in step **a** to the **Tokens** Product. Configure each Feature with the **Execution Count** licensing type. For each Feature, set the number of executions to the number of tokens that the customer should be charged for the use of the application or function.

**NOTE** Executions are not actually consumed from these Features when you execute any of the applications. The Features only serve as a repository for the prices. You must protect your applications with Sentinel LDK Envelope or Sentinel Licensing API using Feature ID 0 or other Features that are not included in the price list.

#### To create the token pool for the customer:

- Define a Feature (to be referred to as the **Token-Pool** Feature) in the Sentinel LDK-EMS catalog.
- Add the **Token-Pool** Feature to the **Tokens** Product. Configure this Feature with the **Execution Count** licensing type. When you deliver the **Tokens** license to the customer, you will increment the number of executions for this Feature by the number of tokens purchased by the customer.

- c. Create an entitlement for the **Tokens** Product. In the entitlement, specify the number of tokens purchased by the customer as described in the previous step.
- d. Deliver the entitlement to the customer. The customer can, for example, install the **Tokens** Product license on a license server machine in their network to allow multiple users to access it.

### Example

The following table provides an example of the Features in the **Tokens** Product.

Feature ID	Description (Application)	Price in Tokens (Number of executions)
101	Spreadsheet application	6
102	Word Processor application	5
103	Presentation Tool application	4
104	Text Editor application	1
105	Scheduler application	3
201	Print function	1
202	Save function	1
9999	Token-Pool (tokens repository)	Number of tokens purchased by the customer (specified in the entitlement)

## 2. Consume Tokens from the Token Pool for Application Usage

Use Sentinel Licensing API to consume the required number of tokens from the token pool when a user executes an application as described below.

### To consume tokens from the token pool:

- a. Upon startup, each application should create a table in memory of the price list for that application, including any functions in the price list that are available in that application. Use the **get\_info** function to load the price list into memory by retrieving the relevant price list Feature IDs and their corresponding prices (number of tokens). In the example above, upon startup, the Word Processor application should load Feature IDs 102, 201 and 202 and their corresponding prices in to the table in memory.
- b. When the requirement for one of the Pricing Feature IDs is encountered in the licensed application, the application should look up the Feature ID in its internal price list table and then use the **login\_scope**



function in the Licensing API to consume the required number of tokens from the token pool. Set the **<execution\_count\_to\_consume>** tag in the scope to the number of tokens to deduct from the pool.

## Locked Licensing Business Models

---

A locked license is limited to usage on a specific machine or by a specific end user.

The locked licensing business models described below are:

- > ["Machine-locked" on the next page](#)
- > ["User-locked" on page 277](#)

## Machine-locked

<b>Sentinel LDK Functionality</b>	Creates an activation key that is locked to a specific machine
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

## Description

The license can only be used on the machine on which it was installed. A machine-locked license can be combined with any of the licensing models in this guide.

### Implementation 1—Locking to a Sentinel SL key

This model is applicable when a Sentinel SL key provides sufficient security for your needs.

1. Select and implement the required licensing models.
2. Distribute your software using a Sentinel SL key. Sentinel SL keys are always locked to a specific machine.

### Implementation 2—Combined locking to both a Sentinel SL key and a Sentinel HL key

This model is applicable when you want to lock your software to a Sentinel HL key for enhanced security, and also wants to use a Sentinel SL key to lock your software to a specific machine. The Sentinel SL key will require remote activation.

1. Select the executable file that you want to license, and determine two Feature IDs by which it will be identified. One Feature ID will be used to lock the license to the Sentinel HL key, and the other to lock the license to the Sentinel SL key and the machine.
2. Select your protection method:
  - *For combined Envelope-based and API-based automatic implementation*  
Protect the executable file using Sentinel LDK Envelope , specifying one of the Feature IDs. In your code, insert a Sentinel Licensing API Login call to other Feature ID.
  - *For API-based automatic implementation*  
In your code, insert Sentinel Licensing API Login calls to both Feature IDs.

3. In Sentinel LDK-EMS, create two Products, one for each Feature ID. Define the license terms for both Products—for example, an execution-based license or a time-limited license.
4. Burn a Sentinel HL key for one of the Products and create a Sentinel SL Product Key for the other Product.
5. Distribute your software with both Sentinel protection keys.

### Implementation 3—Locking to a Sentinel HL key

This model is applicable when you want to lock the license to both a machine and a Sentinel HL key—but for security reasons, the end user will not be able to activate a Sentinel SL key online.

This implementation requires a utility to be written that will collect the required identifiers from the machine before or during installation of your software, and subsequently every time your software is run. The initial identifiers are saved in the read-only memory of the protection key, and the run-time identifiers are written to the read/write memory on the Sentinel HL key and validated against the initial identifiers.

**NOTE** It is recommended that you contact Thales Sentinel Professional Services for a detailed implementation plan.

## User-locked

<b>Sentinel LDK Functionality</b>	Compares end user data saved in the Protection Key memory with a value collected during run-time
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	API-based automatic implementation

## Description

The license can only be run by a specific logged-in end user. A user-locked license ensures that only an entitled end user can activate your software. This model can be particularly useful when your software resides on a server, or is activated by a remote end user. A user-locked license can be combined with any of the licensing models in this guide.

## Implementation

Select and implement the required licensing model, and distribute your software with the appropriate Sentinel protection key programmed with the license.

There are two ways to lock the key to a specific end user:

### > Option 1: Predefined locking

Identification is based on the login user name defined in the operating system. Predefined locking enables a number of authorized end users to access your software residing on a single machine.

- a. When a license is purchased, request the login user name of the end user for whom the license is intended.
- b. Use Sentinel LDK-EMS to save the user name to the Read-Only memory of a Sentinel protection key.
- c. During run-time, read the user name from the machine, and use the Sentinel Licensing API to validate it against the user name saved on the Sentinel protection key.

### > Option 2: Password locking

During installation, the end user defines a user name and password, which are later required in order to log in to your software. Password locking is less convenient for an end user, but provides extra security. When a Sentinel HL key is used, your software can be installed on more than one computer, but can only be accessed when the Sentinel HL key is connected.

- a. During installation, request the end user to define a user name and password.
- b. Use the Sentinel Licensing API to save the data to the Read/Write memory on the Sentinel HL key.

- c. During run-time, require the end user to log in, and validate the user name and password against the data saved on the Sentinel protection key.

## Mobile Licensing Business Models

---

Many software vendors are looking for ways in which they can accommodate the growing trend towards a mobile workforce. The models in this section provide options for mobile licenses.

The models described below are:

- > ["Portable" on the next page](#)
- > ["Commuter" on page 281](#)
- > ["Software on a Key" on page 282](#)

## Portable

<b>Sentinel LDK Functionality</b>	Locks the license to a hardware-based Sentinel HL key
<b>Software Distribution Method</b>	Physical package
<b>Applicable Key Types</b>	All Sentinel HL keys
<b>Protection Method</b>	<ul style="list-style-type: none"><li>&gt; Envelope-based automatic implementation</li><li>&gt; API-based automatic implementation</li></ul>

## Description

Your software can be installed on any number of machines, providing flexibility, but can only run on the machine to which the Sentinel HL key is connected.

## Implementation

1. Select and implement the required licensing business model.
2. Distribute your software with the appropriate Sentinel HL key, programmed with the license.



## Commuter

<b>Sentinel LDK Functionality</b>	Enables a network-based license to be detached to a separate machine while locked to a Sentinel SL key
<b>Software Distribution Method</b>	Electronic distribution
<b>Applicable Key Types</b>	Sentinel SL
<b>Protection Method</b>	<ul style="list-style-type: none"><li>&gt; Envelope-based automatic implementation</li><li>&gt; API-based automatic implementation</li></ul>

## Description

A license can be temporarily detached from a network pool—using Sentinel Admin Control Center—to enable off-line use of your software. For example, when employees leave the office to work off site, they can take their laptops with them and continue using the protected software locally.

## Implementation

1. Select and implement the network concurrency licensing model, ensuring that the license can be locked to a Sentinel SL key and that detachable licenses are enabled.
2. Distribute your software with a Sentinel SL key, ensuring that the system administrator at your end-user site knows how to permit and manage detachable licenses.
3. If the employee requires the detached license for less time than originally planned, the license can be manually returned to the network pool before its expiration date.

## Software on a Key

<b>Sentinel LDK Functionality</b>	Locks the license to a Sentinel HL Drive microSD key that also contains your software
<b>Software Distribution Method</b>	Physical package
<b>Applicable Key Types</b>	Sentinel HL Drive microSD
<b>Protection Method</b>	<ul style="list-style-type: none"><li>&gt; Envelope-based automatic implementation</li><li>&gt; API-based automatic implementation</li></ul>

## Description

Both your software and the license are stored on a Sentinel HL Drive microSD key, providing maximal mobility. The Sentinel HL Drive key contains up to 64 GB of flash memory on a microSD card in addition to the license data memory, enabling all of your software to reside on the key. This method is applicable for software that can be run from an external key without necessitating installation on a hard disk.

This method can be applied to all licensing business models for which a hardware-based key is used.

## Implementation

1. Select and implement the required licensing business model.
2. Distribute your software on a Sentinel HL Drive microSD key, together with the software's license.

## Network Licensing Business Models

---

Network licenses are designed for a network environment, in which the vendor's software is run by multiple end users or on multiple workstations. In such an environment, a single Sentinel protection key can be used to protect and monitor usage of the vendor's software across the network. Network licenses can be implemented in conjunction with other licensing business models such as component-based or metering. A network license can be concurrency-based, site-specific, or both.

The network licensing business models described below are:

- > ["Limited Concurrent End Users in a Network" on the next page](#)
- > ["Time-limited Concurrent End Users in a Network" on page 286](#)
- > ["Execution-limited Concurrent End Users in a Network" on page 288](#)
- > ["Volume" on page 290](#)
- > ["Site" on page 291](#)

## Limited Concurrent End Users in a Network

<b>Sentinel LDK Functionality</b>	Manages the number of concurrent software end users
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL (Driverless configuration) keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

### Description

A concurrency-limited network license limits the number of end users concurrently accessing the licensed application in a network environment, preventing additional activations and unintentional piracy if the maximum number of allowed concurrent licenses has been reached. The same license can be used by more than one end user or workstation, so long as the total number of users remains within the concurrency limit.

Sentinel Admin Control Center provides the end users' system administrator with the tools to track license users, and to terminate an inactive session.

### Implementation

1. Select the executable file that you want to license, and determine by which Feature ID the file or function will be identified.
2. Select your protection method:
  - *Envelope-based automatic implementation*  
Protect the executable file using Sentinel LDK Envelope, specifying its Feature ID.
  - *API-based automatic implementation*  
In your code, insert a Sentinel Licensing API Login call to the Feature ID.
3. In Sentinel LDK-EMS:
  - a. Create a Product that includes the Feature ID, and define the license type as Perpetual.
  - b. Set the concurrency counter to the required maximum number of concurrent licenses, and determine whether concurrent instances will be counted for each station, each login or each process.

**TIP** You can specify the number and type of concurrent instances each time a specific order is created. This enables you to use the same Product to produce more than one license, each with a different number of seats.

4. Distribute your software with a Sentinel protection key programmed with the license.

## Time-limited Concurrent End Users in a Network

<b>Sentinel LDK Functionality</b>	Manages the number of concurrent software end users in a network and the time period over which your software can be used
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; Sentinel HL NetTime</li> <li>&gt; All Sentinel HL (Driverless configuration) keys except HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

### Description

A combined concurrency-limited and time-limited network license restricts both the number of end users concurrently accessing the licensed application in a network environment and the period during which the license is valid. The same license can be used by more than one end user or machine, so long as the total number of users remains within the concurrency limit.

Sentinel Admin Control Center provides the end user's system administrator with the tools to track license users, and to terminate an unused session.

### Implementation

1. Select the executable file that you want to license, and determine by which Feature ID it will be identified.
2. Select your protection method:
  - *Envelope-based automatic implementation*  
Protect the executable file using Sentinel LDK Envelope, specifying its Feature ID.
  - *API-based automatic implementation*  
In your code, insert a Sentinel Licensing API Login call to the Feature ID.
3. In Sentinel LDK-EMS:
  - a. Create a Product that includes the Feature ID, and define the expiration date or number of days until expiration.
  - b. Set the concurrency counter to the required maximum number of concurrent licenses, and determine whether concurrent instances will be counted for each station, each login or each process.

**TIP** You can specify the number and type of concurrent instances each time a specific order is created. This enables you to use the same Product to produce more than one license, each with a different number of seats.

4. Distribute your software with the appropriate network-based Sentinel protection key programmed with the license.

## Execution-limited Concurrent End Users in a Network

<b>Sentinel LDK Functionality</b>	Manages the number of concurrent software end users in a network and the maximum number of software executions
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL (Driverless configuration) keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

### Description

A combined concurrency- and execution-limited network license restricts both the number of end users concurrently accessing the licensed application in a network environment and the total number of executions for each license. The same license can be used by more than one end user or machine, so long as the total number of users remains within the concurrency limit. The number of executions is calculated across the network, regardless of which end user runs your software or on which machine it is run.

Sentinel Admin Control Center provides the end users' system administrator with the tools to track license users, and to terminate an unused session.

### Implementation

1. Select the executable file or software function that you want to license, and determine by which Feature ID the file or function will be identified.
2. Select your protection method:
  - *Envelope-based automatic implementation*  
Protect the executable file using Sentinel LDK Envelope, specifying its Feature ID.
  - *API-based automatic implementation*  
In your code, insert a Sentinel Licensing API Login call to the Feature ID.

**NOTE** If your protection method is feature-based, apply API-based automatic implementation; if your protection method is for each executable file, you can apply either Sentinel LDK Envelope-based or Sentinel Licensing API-based automatic implementation.

3. In Sentinel LDK-EMS:



- a.** Create a Product that includes the Feature ID, and define the maximum number of executions.
  - b.** Set the concurrency counter to the required number of concurrent licenses, and determine whether the concurrent instances will be counted for each station, each login or each process.
- 4.** Distribute your software with the appropriate network-based Sentinel protection key programmed with the license.

## Volume

<b>Sentinel LDK Functionality</b>	Enables a network-based license to be detached to a separate machine while locked to a Sentinel SL key
<b>Software Distribution Method</b>	Electronic distribution
<b>Applicable Key Types</b>	Sentinel SL (including CL)
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

## Description

A volume license enables you to sell a pool of licenses to an organization, without requiring product activation on every machine, while still enforcing the maximum number of installed workstations.

A license can be temporarily detached from the network pool to enable off-line use of your software. In this case, a client machine periodically detaches a time-limited license at predefined intervals—transparently to the end user. The license is installed locally and remains usable even if the network connectivity is lost, as long as the detachment is still valid.

## Implementation

1. Select the executable file that you want to license, and determine by which Feature ID it will be identified.
2. Select your protection method:
  - *Envelope-based automatic implementation*  
Protect the executable file using Sentinel LDK Envelope, specifying its Feature ID.
  - *API-based automatic implementation*  
In your code, insert a Sentinel Licensing API Login call to the Feature ID.
3. In Sentinel LDK-EMS, create a Product that contains the Feature ID used in the protection phase of the implementation. Ensure that the license terms enable network concurrency, locking to a Sentinel SL key, and detachable licenses.
4. Distribute your software with a Sentinel SL key for network use, ensuring that the system administrator at your end-user site knows how to permit and manage detachable licenses.
5. Using the Sentinel Licensing API, implement the license's detachment in the protected application. You may wish to let the customer organization decide the detached license period and renewal intervals.

## Site

<b>Sentinel LDK Functionality</b>	Locks the license to a specific domain, network, or subnet
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	API-based automatic implementation

## Description

A site license is a license that is locked to a specific domain, network, or subnet. A site license can be combined with any of the licensing business models in this guide.

## Implementation

1. Select and implement the required licensing business model.
2. Envelope your software for additional security (optional).
3. Distribute your software using the appropriate Sentinel protection key.
4. To lock the Sentinel protection key to the license, collect the site identifier (domain, subnet or network) from the customer. An identification value is written to the Sentinel protection key. The application then validates the identifier every time your software runs.
5. There are two ways in which you can collect site-specific data and save it on the Sentinel protection key:

- *Option 1: Site identifier collected prior to installation*

Provides more security, but is less convenient for the customer.

When a license is purchased, send the customer a utility that collects the required site identifier from the customer.

Use Sentinel LDK-EMS to save the identification value to the Read-Only memory of the Sentinel protection key.

- *Option 2: Site identifier collected during installation*

Requires less interaction with the customer, but is less secure.

During installation, collect the site identifier from the machine on which your software is installed.

Use the Sentinel Licensing API to verify that there is no existing site identifier saved in the Read/Write memory on the Sentinel protection key.

If the memory does not contain an existing site identifier, save the value to the Read/Write memory on the Sentinel protection key.

6. During run-time, read the site identifier, and use Sentinel Licensing API to validate it against the identification value saved on the Sentinel protection key.

## Sales Boosting Licensing Business Models

---

The sales boosting licensing business models described below are:

- > ["KickStart \(Quick-delivery Grace\)" on the next page](#)
- > ["Referral-based Sales" on page 296](#)
- > ["Automatic Sales Agent" on page 298](#)

## KickStart (Quick-delivery Grace)

<b>Sentinel LDK Functionality</b>	Grants a grace period to use software until key is delivered
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

## Description

Locking a license to a Sentinel HL key provides a higher level of security than locking to a Sentinel SL key, but delivery of the Sentinel HL key to an end user can take time. This model enables you to electronically supply your software with a quick-delivery license locked to a Sentinel SL (software) key (“KickStart license”) as soon as an order is processed. For increased protection, you may choose to limit some software functions in the KickStart license.

The KickStart license can be used as part of a two-phased sales model:

- > **Phase 1:** The end user purchases your software, and a 30-day KickStart license with limited functionality is supplied electronically.

**NOTE** The KickStart license is typically defined for a period of up to 90 days.

- > **Phase 2:** The Sentinel HL key, programmed with the full license (the “final” license), is delivered within 30 days. The end user replaces the KickStart license with the full license, using the RUS utility.

The KickStart license also serves as a super-distribution mechanism, since it will run for the grace period on any computer on which it is installed.

## Implementation

1. Determine which global Feature ID you will use for the KickStart license.
2. Select the software functions that you want to include only in the full license, and determine by which Feature IDs each function will be identified.
3. Select a protection method and do one of the following:

**For Envelope-based automatic implementation:**

1. Determine which global Feature ID you will use for the full license.
2. Create two executable files, one with limited functionality for the KickStart license, and the other with full functionality for the full license.
3. Envelope each executable file separately, using the global Feature IDs you defined for the KickStart and full licenses respectively.

**For API-based automatic implementation:**

1. In your code, insert a Sentinel Licensing API Login call to the global Feature ID for the KickStart license.
2. In your code, for each software function you want to include only in the full license, insert Sentinel Licensing API Login calls to the appropriate Feature IDs.

**In Sentinel LDK-EMS:**

1. Create a Product that includes the global Feature ID for the KickStart license.
2. Select the Trialware/Unlocked Product attribute.
3. Distribute your software with Sentinel LDK Run-time Environment. Your software can run for a grace period of 30 days and can be installed on any other computer, for a 30-day period, as a super-distribution mechanism.
4. Create a Product that includes the full license Feature IDs.
5. Define appropriate license terms for each Feature.

**NOTE** If the full license is based on a metered licensing model, metering will commence only when the full license is activated and not during the grace period.

6. Distribute your software with a Sentinel protection key programmed with the full license.

## Referral-based Sales

<b>Sentinel LDK Functionality</b>	Creates an unlocked trialware Product that allows for unrestricted distribution of the protected software
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL keys</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

## Description

A bonus mechanism that encourages end users to serve as “promoters” for software they find useful. When an end user refers software to someone and a purchase is made based on that referral, you give a bonus to the referrer.

This model requires the creation of two vendor mechanisms:

- > **User data collection mechanism**—You maintain an end-user database in which registered software owners (*referrers*) are linked to potential users to whom the software was referred (*referees*). Data for the database can be sent to you by either the referrer or the referee, using a variety of data collection mechanisms. For example, data can be collected via a form displayed during software activation or on a Web site.
- > **Bonus-granting mechanism**—When the software is purchased, your end-user database is queried. If the purchase was made as the result of a referral, the referrer receives a bonus from you.

The following implementation guidelines describe how to set up the referral-based sales model, based on:

- > Using trialware as the evaluation mechanism.
- > Distributing the purchased software with a software-based Sentinel SL key.
- > Collecting information from the referee during software activation.

## Implementation

1. Create a trialware version of your software.
2. End users who have already purchased your software send the trialware to other potential users.
3. When a new user purchases your software—as part of your software activation process using Sentinel LDK functionality—prompt the new user to provide you with the name and contact information of the end user who referred your software to them.



4. Reward the referrer.

**NOTE** This is a typical implementation. However, the referral-based sales model can also be applied to other licensing business models, including those models that use a hardware-based Sentinel HL key.

## Automatic Sales Agent

<b>Sentinel LDK Functionality</b>	Manages module usage
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL keys except Sentinel HL Basic</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	API-based automatic implementation

## Description

When an end user purchases a subset of software modules, the sales staff is often requested to follow up the purchase and to interest the user in additional modules. With Sentinel LDK, your software can serve as its own automatic sales agent, providing the end user with the ability to work with additional modules and encouraging purchase of any modules that are identified as being of interest to the end user. This model consists of a number of phases:

- > *Phase 1:* The end user purchases a subset of software modules. You supply a license that includes the option to install additional bonus modules so that the user can experiment with them.
- > *Phase 2:* The end user uses your software, including the bonus modules. Behind the scenes, your software monitors and evaluates usage of the bonus modules.
- > *Phase 3:* Once the usage threshold of a monitored module has been reached, the module is considered “of value” and Sentinel LDK progressively restricts usage of that module. Concurrently, the Automatic Sales Agent comes into effect, issuing pop-up messages encouraging the end user to purchase the module.
- > *Phase 4:* When an end user purchases a license for an additional module, the license is seamlessly upgraded at the end-user site, using the RUS utility, and the relevant bonus modules are changed to fully-licensed modules.

## Implementation

1. Determine which Feature ID you will use for global protection of your software.
2. Select the modules that you want to license separately, and determine by which Feature ID each of the modules will be identified.
3. In your code, insert Sentinel Licensing API Login calls to all Feature IDs.
4. In Sentinel LDK-EMS, create a Product that includes only the global software Feature ID and define the license terms.
5. Determine the parameters for gauging module usage, and define them in your code, for example:
  - The number of times a monitored module has been activated during a time period

- The accumulated usage time of a monitored module
  - The number of clicks on an item in the user interface
6. In Sentinel LDK-EMS, in the Protection Key memory, define the usage threshold.
  7. Envelope your software for additional security (optional).
  8. Distribute your software with the appropriate Sentinel protection key programmed with the license for the initial purchase, not including licenses for the bonus modules.
  9. Using the Sentinel Licensing API:
    - a. Calculate the accumulated usage of the gauging parameters.
    - b. Write the result to the Protection Key memory.
    - c. Compare the accumulated usage with the defined threshold.

When usage of a bonus module passes the threshold, begin to implement the restrictions, for example:

- Progressively slow down the speed of the module as the time passes or as usage increases
  - Progressively increase the number of Automatic Sales Agent pop-up messages as the time passes or as usage increases
  - Prevent the module from saving a snapshot of work that has been done
10. In Sentinel LDK-EMS, create a Product that includes both the global software Feature ID and the Feature ID for the module identified as being sellable, and define the license terms.
  11. When the end user decides to purchase a license for a bonus module, update the license on the Sentinel protection key to include the purchased module, using the RUS utility.

## Perpetual Licensing Business Models

---

The perpetual licensing business models described below are:

- > ["Standard Perpetual" on the next page](#)
- > ["Perpetual Unlocked" on page 302](#)

## Standard Perpetual

<b>Sentinel LDK Functionality</b>	Creates an unconditional license
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; All Sentinel HL keys</li> <li>&gt; Sentinel SL (including CL)</li> </ul>
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

## Description

The traditional perpetual, unlimited licensing model can serve as a basis for other, more creative marketing strategies, for example:

- > Your software is initially supplied with a perpetual license. The end user purchases additional modules as required.
- > The initial release is supplied with a perpetual license. You plan to implement more sophisticated licensing models with future releases.
- > A limited license (“bronze”) is converted to a perpetual license (“gold”) for additional payment.

## Implementation

1. Select the executable file that you want to license, and determine by which Feature ID it will be identified.
2. Select your protection method:
  - *Envelope-based automatic implementation*  
Protect the executable file using Sentinel LDK Envelope, specifying its Feature ID.
  - *API-based automatic implementation*  
In your code, insert a Sentinel Licensing API Login call to the Feature ID.
3. In Sentinel LDK-EMS, create a Product that includes the Feature ID and define a perpetual license for the Feature.
4. Use the RUS utility to update a license currently held by the end user with the new license.

## Perpetual Unlocked

<b>Sentinel LDK Functionality</b>	Creates an unconditional unlocked license
<b>Software Distribution Method</b>	<ul style="list-style-type: none"> <li>&gt; Physical package</li> <li>&gt; Electronic distribution</li> </ul>
<b>Applicable Key Types</b>	<ul style="list-style-type: none"> <li>&gt; None</li> </ul>
<b>Protection Method</b>	<ul style="list-style-type: none"> <li>&gt; Envelope-based automatic implementation</li> <li>&gt; API-based automatic implementation</li> </ul>

## Description

When using an unlocked license, the application is protected against disassembly and modification, but the license is not locked to a specific computer, and no licensing restrictions are applied.

Unlocked licenses are applicable for any of the following situations:

- > You want to distribute the software as an Unlocked Product with no time limit (or with an extended time limit). For example, you may want to allow users to access basic functionality as long as they want, with the option to buy an upgrade later to access advanced functionality.
- > You want to use a licensing system other than Sentinel LDK.
- > Licensing is not an issue. For example, you are distributing medical equipment with embedded software. Since the software is specific to your equipment, you are not concerned about the possibility of duplication of the software.

## Implementation

1. Select the executable file that you want to license, and determine by which Feature ID it will be identified.
2. Select your protection method:
  - *Envelope-based automatic implementation*  
Protect the executable file using Sentinel LDK Envelope, specifying its Feature ID.
  - *API-based automatic implementation*  
In your code, insert a Sentinel Licensing API Login call to the Feature ID.
3. In Sentinel LDK-EMS, create an Unlocked Product (Perpetual) that includes the Feature IDs that you want to include in the unlocked license.

## Part 6 - Appendices

### In this section:

---

- > ["Understanding the Sentinel LDK Master License" on page 305](#) – Describes the modules that are available to software vendors in the Sentinel LDK Master license.
- > ["Sentinel LDK Run-time Network Activity" on page 319](#) – Describes the type of network activity that occurs in the communication between the Sentinel License Manager and a protected application, and between the local Sentinel License Manager and remote Sentinel License Managers.
- > ["Maximum Number of Features in a Sentinel HL Key" on page 322](#) – Describes considerations that determine the maximum number of Features that can be contained in a Sentinel HL key.
- > ["How Sentinel LDK Detects Machine Cloning" on page 323](#) – Describes the techniques employed by Sentinel LDK to prevent unauthorized use of protected software when the virtual machine on which the software is installed is cloned.
- > ["How Sentinel LDK Protects Time-based Licenses With V-Clock" on page 346](#) – Describes the technology used in Sentinel LDK to prevent a user from extending the duration of a software license that is locked to a Sentinel SL key.
- > ["How to Optimize Performance for Sentinel LDK Run-time Environment" on page 349](#) – Describes how you can optimize the performance in the Sentinel LDKRun-time Environment
- > ["Upgrading Sentinel HL Keys" on page 350](#) – Describes how you can upgrade a Sentinel HL (HASP configuration) key to a Sentinel HL (Driverless configuration) key or convert a Sentinel HL standalone key to a network key.
- > ["Protecting Applications in Docker Containers" on page 355](#) – Describes how to protect applications that execute in a Docker container.
- > ["Protecting Applications in Linux LXC Containers" on page 362](#) – Describes how to protect applications that execute in an LXC container under Linux.

- > ["Troubleshooting" on page 366](#) – Provides a checklist to help you solve some of the most common problems that your customers might encounter when using the Sentinel HL keys. Also includes a list of specific problems you or your customers may experience, together with the solutions.
- > – Provides a comparison of the functionality between Sentinel LDK-EMS and Sentinel EMS. This appendix assists vendors who want to choose between these two products or who are considering migrating from Sentinel LDK-EMS to Sentinel EMS.



# APPENDIX A: Understanding the Sentinel LDK Master License

This appendix describes the Sentinel LDK Master license. Its purpose is to assist you in understanding how your Sentinel LDK Master license from Thales is implemented, and how to make decisions about your license update requirements.

*In this appendix:*

- > ["What is the Master License?" on the next page](#)
- > ["Where is the Master License Located?" on the next page](#)
- > ["Updating the Master License" on page 307](#)
- > ["Modules Summary" on page 307](#)
- > ["Trial Licenses Provided With Sentinel LDK" on page 308](#)
- > ["Licensing Concepts" on page 309](#)
- > ["Product Activation Module" on page 310](#)
- > ["New SL Key Pool" on page 311](#)
- > ["Network Seats" on page 311](#)
- > ["Unlocked Trialware Module" on page 315](#)
- > ["Unlocked Unlimited Module" on page 315](#)
- > ["V-Clock Module" on page 316](#)
- > ["AppOnChip Module" on page 316](#)
- > ["Channel Partner Module" on page 317](#)
- > ["Reporting Module" on page 317](#)
- > ["Cloud Licensing Module" on page 317](#)

## What is the Master License?

The Sentinel LDK Master license entitles you to work with Sentinel LDK and all of its components, and to use Sentinel LDK to protect and license your applications.

The Master license provide access to the basic functionality of Sentinel LDK. The Master license also contain the following types of modules:

### > Subscription

Sentinel LDK provides optional advanced functionality that adds value, simplifies operation, and increases security for your applications.

Subscription modules grant you permission to use this advanced functionality for a period of time that is defined in your subscription to each module.

Most Subscription modules are stored in the Sentinel LDK Master license. Modules that provide advanced protection functionality for your applications are stored externally as described in "[Where is the Master License Located?](#)" [below](#)

### > Meters

For certain functions (such as creation of SL keys), you are charged per usage. For each function, you purchase a pool of units, which are added to a meter for the function. A unit is subtracted from the meter for each usage of the function. When the meter is close to depletion, you must purchase a new supply units to refill the meter. You can also subscribe to access for any of these functions and receive an unlimited supply for the function for the duration of the subscription.

## Where is the Master License Located?

For Sentinel LDK-EMS hosted by Thales, the Sentinel LDK Master license is located on the Thales server where your instance of Sentinel LDK-EMS is installed.

For Sentinel LDK-EMS on-premises, The Sentinel LDK Master license is located on your Master key. To generate licenses, the Master key must be connected at all times to the machine where Sentinel LDK-EMS is installed. (For more information on the Master key, see "[Sentinel LDK Vendor Keys](#)" on page 22.)

(If you use Sentinel License Generation API to generate licenses, the Master key must be connected at all times to the machine where the License Generation API runs.)

Modules that relates to protection of your application are located on your Developer key. This key must be connected to the machine where Sentinel LDK Envelope runs.

You can view the modules on your Developer key in Sentinel Admin Control Center.

**NOTE** The Developer key and Master key can be accessed using a remote connection. For more information, see the [Sentinel LDK Installation Guide](#).

## Updating the Master License

For Sentinel LDK-EMS hosted by Thales, all maintenance of the Sentinel LDK Master license is handled by Thales. If you want to refill the meter for a metered function or add a new module, you simply place an order with your Thales customer representative.

For Sentinel LDK-EMS on-premises, you maintain your Sentinel LDK Master license using Sentinel LDK-EMS. You refill meters or add subscription modules to your Master key as described in ["Maintaining Your Sentinel LDK Master Key" on page 144](#).

You can view the modules in the Sentinel LDK Master license in Sentinel LDK-EMS (from **Administration > Master**). You can also use Sentinel Admin Control Center to view all the modules in your Master license or Developer key.

To add modules to your Developer key, you can use the Sentinel Master Key Update Utility:

1. From the Start menu, open **Sentinel LDK**. From the Sentinel LDK Launcher, select **Additional Tools > Sentinel Master Key Update Utility**.
2. Use the utility to generate a C2V file for your Developer key.
3. Place your order and send the C2V file to Thales.
4. Thales returns a V2C file to you.
5. Use the utility to apply the V2C file to your Developer key.

## Modules Summary

This section summarizes the modules that are available for the Sentinel LDK Master license. The modules that you purchase or subscribe to depend on your specific requirements.

To see information about the modules in your Sentinel LDK Master license: In Sentinel LDK-EMS, select **Administration > Master**.

To see information about the modules on your Developer key, start Admin Control Center on the machine where the Developer key is connected. In the navigation pane, click **Sentinel Keys**. In the entry for the Developer key, click **Features**.

The following table lists the modules that can be included in a Sentinel LDK Master license or on a Developer key. The table includes the Feature ID, which is sometimes used to refer to the module.

Module	Used For	Value	Location	Feature ID
<a href="#">AppOnChip Module</a>	Advanced protection	Subscription expiration date	Developer key or Master License	10
<a href="#">Channel Partner Module</a>	Advanced channel partner functionality in Sentinel LDK-EMS	Subscription expiration date	Master License	12
<a href="#">Cloud Licensing Module</a>	License generation	Subscription expiration date	Master License	19
<b>Network Seats</b>	License generation	Number of seats or subscription expiration date	Master License	
<a href="#">HL Pool of Seats</a>				14
<a href="#">SL Pool of Seats</a>				4
<a href="#">Value of Unlimited Seats</a>		Fixed value		5
<a href="#">New SL Key Pool</a>	License generation	Number of keys or subscription expiration date	Master License	3
<a href="#">Product Activation Module</a>	Sentinel SL protection key activation using Sentinel LDK-EMS	Subscription expiration date	Master License	2
<a href="#">Reporting Module</a>	Sentinel LDK-EMS report generation	Subscription expiration date	Master License	6
<a href="#">Unlocked Trialware Module</a>	License generation	Subscription expiration date	Master License	7
<a href="#">Unlocked Unlimited Module</a>	License generation	Subscription expiration date	Master License	8
<a href="#">V-Clock Module</a>	License generation	Subscription expiration date	Master License	9

## Trial Licenses Provided With Sentinel LDK

First-time users of Sentinel LDK receive the following trial licenses as part of the Sentinel LDK package:

Product Activation module	30 days
Reporting module	30 days
V-Clock module	30 days
AppOnChip module	30 days
Channel Partner module	30 days
New SL Key Pool	15 units
SL Pool of Seats	15 units
HL Pool of Seats	15 units
Cloud Licensing module	15 units. See <a href="#">"Trial for Cloud Licensing" below</a> .

These are fully-functional licenses for Sentinel LDK, provided for the duration or number of units listed above. Each module is described in this appendix.

## Trial for Cloud Licensing

The Cloud Licensing module is a subscription module that is usually set to an expiration date when you subscribe to this feature. However, for first-time users of Sentinel LDK who have not subscribed to cloud licensing, the Cloud Licensing module is set to allow a specific number of activations for cloud-enabled SL licenses.

### NOTE

- > To generate cloud-enabled SL licenses, you must first set the **Cloud Licensing** configuration parameter in Sentinel LDK-EMS to **Enabled**. Once cloud licensing is enabled, each activation of an SL key (whether or not intended for cloud licensing) will consume a unit from the Cloud Licensing module.
- > When all of the units in the Cloud Licensing module have been consumed: If you have not purchased a subscription for the Cloud Licensing module, you must change the **Cloud Licensing** configuration parameter to **Leave as is**. Otherwise, Sentinel LDK-EMS will generate an error when you attempt to activate an SL license.

## Licensing Concepts

In the descriptions of the Sentinel LDK Master license model, the following concepts are used:

- > **Unlocked Trialware Product:** A Product that can be used as trialware, or during a grace period. Unlocked trialware Products are not locked to a specific machine and do not require activation for a limited period. Unlocked trialware Products typically have a duration of 30 to 90 days or 30 executions. This period can be

set to begin either from the date of first use of the application or from the date that the license was generated. (The Unlocked trialware Product was formerly referred to as a *provisional Product*.)

- > **Unlocked Unlimited Product:** A Product that does not lock a protected application to a specific machine and does not necessarily impose any licensing restrictions on the use of the protected application. The Product can be granted a perpetual license or can be limited to any length of time that you choose. This enables the vendor to use Sentinel LDK to protect the application, but use a different mechanism to license the application (or impose no license restrictions on the application).
- > **Activation:** The process in which an SL key is locked to a specific computer. Following Activation, the protected software can be used on the end user's computer according to the license that was installed during the Activation process.

For the list of Sentinel LDK license types, see ["Assigning License Terms to Features" on page 104](#).

- > **Concurrency:** A licensing attribute that can be specified to allow a single protection key on a computer in a network to be used by one or more instances of a protected application running on different computers in the network.

Concurrency is defined separately for each Feature in a Product.

Each instance of the protected application that can be used simultaneously is referred to as a *network seat* (or a *floating license*).

Network seats are not assigned to specific users. Instead, the concurrency attributes specify how many instances (network seats) of the Feature in protected application can be used simultaneously within the customer's network. The customer purchases a specific number (or an unlimited number) of network seats.

For example: A customer purchases 10 network seats for the *Basic* Feature and 5 network seats for the *Advanced Tools* Feature for a protected application. As a result, 10 end users can run the application and use the *Basic* Feature simultaneously. 5 of these users can also use the *Advanced Tools* Feature simultaneously. All the users must be part of the network where the protection key is located.

Management of the license in the network is controlled using the Sentinel License Manager.

For more information about concurrency, see ["Specifying the License Terms for Features in a Product" on page 110](#).

---

## Product Activation Module

Sentinel LDK provides a mechanism to easily perform interactive license updates on an end user's machine. This is accomplished by generating a Product Key for an entitlement in Sentinel LDK-EMS and providing this code to the end user. The end user accesses the Sentinel LDK-EMS Customer Portal over the Internet and enters the Product Key. Sentinel LDK-EMS then retrieves the necessary information about the end user's machine or existing license and completes the process to update the license on the user's machine. (This process can also be accomplished in program code using Sentinel LDK-EMS Web Services.)

This mechanism is typically used to activate an application on the end user's machine (that is, to lock an SL key for the application to the machine), although the mechanism can be used for other types of license updates.

To use the Product Key mechanism to update an SL key, you must have the Product Activation module in your Sentinel LDK Master license. The Product Activation module is either perpetual or issued for a limited time period. This depends on your purchase plan or subscription plan for Sentinel LDK. For more information, consult with your Thales sales representative.

The Product Activation module is not required if you only want to use the Product Key mechanism to update HL keys.

## New SL Key Pool

---

Each time a new SL key is created for a given machine at a customer site, an SL key unit is consumed from the New SL Key Pool in the Sentinel LDK Master license.

A new SL key is created in these situations:

- > An end user submits a Product Key for your software for the first time for a given machine. End users can submit a Product Key online, or they can request and receive an activation file to apply manually.
- > You use Sentinel License Generation API to generate a license code for the first time for a given machine.

To create new SL keys, you may need to purchase a pool of SL key units. (This depends on your purchase plan or subscription plan for Sentinel LDK.)

When the New SL Key Pool is low, you purchase additional SL key units (if required by your plan). You can configure Sentinel LDK-EMS to send notifications when the pool reaches a predefined threshold, to ensure that you never run out of SL key licenses for your software. For additional information about configuring notifications, see the [Sentinel LDK-EMS User Guide](#).

When you purchase SL key units, Thales adds an extra 10% to the number of units provided, to compensate for situations in which an SL key unit should not have been deducted from your Sentinel LDK Master license. (For example, if a customer's hard disk drive fails and the customer must reinstall the software on a new disk drive or a different computer, you may choose to provide an additional activation even though the customer did not purchase a second license.)

If there are no SL key units remaining in your Sentinel LDK Master license (and your purchase plan or subscription plan requires that you purchase SL key units), you will not be able to perform an activation that installs a new SL key on a machine.

## Network Seats

---

Network seats are required to enable users to run your software concurrently in a network environment when your Product is licensed with a Sentinel SL key or Sentinel HL concurrency-enabled key. (Network seats from

your Sentinel LDK Master license are not required when your Product is licensed with a Sentinel HL Net or NetTime key.) When you enter an order for your customer: For each Feature in the Product, you specify whether concurrency is enabled for that Feature, and the number of instances (network seats) that are supported.

Your Sentinel LDK Master license contains the pools of network seats described below. To enable concurrency for Features, you may need to purchase network seat units for the appropriate pool in your Sentinel LDK Master license (if required by your purchase plan or subscription plan).

#### > **SL Pool of Seats**

Each time a customer activates your software, the number of concurrent instances that you included in the Product is deducted from the SL Pool of Seats on your Sentinel LDK Master license.

#### > **HL Pool of Seats**

Each time you burn or update an HL key for a Product with concurrency, the number of network seats that you add to the key is deducted from the HL Pool of Seats on your Sentinel LDK Master license.

If a Product contains a number of Features that have different concurrency attributes, and the number of network seats that are provided for the Features differs, the total number of seats deducted from your Sentinel LDK Master license is that of the Feature with the highest number of seats.

When the number of network seats remaining in the relevant pool on your Sentinel LDK Master license is low, you replenish it by purchasing additional network seats (if required by your plan). You can configure Sentinel LDK-EMS to send you notification when the number of seats remaining reaches a predefined threshold, to ensure that you never run out of network seats for your software.

You do not require network seats on your Sentinel LDK Master license if you do not intend to enable concurrency.

## How New Activations and Update of Your Software Affect the Pool

When your protected application is first activated at the customer site or when you burn an HL key for a Product with concurrency, Sentinel LDK examines which Feature in the Product contains the greatest number of concurrency instances. The number of concurrent instances defined in that Feature is deducted from the SL or HL pool of seats. (The concurrency in all other Features is ignored.)

For the Sample Product in the graph below, the customer purchased as follows:

- > For the Print Feature: 12 network seats
- > For the Save Feature: 5 network seats
- > For the Export Feature: 6 network seats

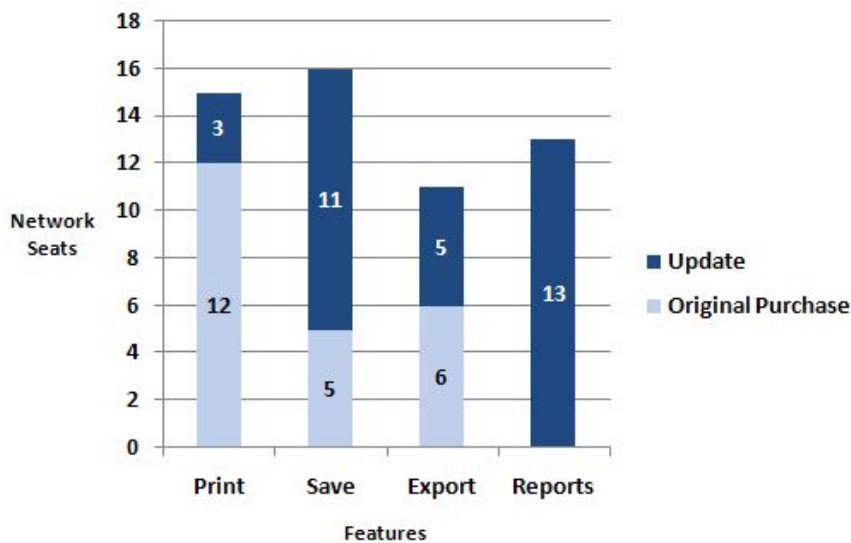
The Print Feature has the greatest number of concurrent instances. Therefore, when the Product is activated, 12 network seats are deducted from the pool.



Later, the customer decided to purchase additional network seats or additional Features in the protected application. For the sample Product in the graph below, the customer purchased as follows:

- > For the **Print** Feature: 3 network seats
- > For the **Save** Feature: 11 network seats
- > For the **Export** Feature: 5 network seats
- > For the **Reports** Feature: 13 network seats

#### Sample Product - Number of Network Seats for Each Feature



When you fulfill the order, Sentinel LDK calculates the number of seats to deduct from your pool of seats as follows:

1. Sentinel LDK determines which Feature had the greatest number of seats until now—in this case, the **Print** Feature with 12 seats.
2. The number of additional seats required for each Feature for the update order is added to the original number of seats that the customer purchased. The chart above indicates the total number of seats that the customer now has.
3. Sentinel LDK determines which Feature now has the greatest number of seats—in this case, the **Save** Feature with 16 seats.
4. The number of seats for the **Print** Feature that the customer had already purchased is deducted from the new total number of seats for the **Save** Feature (16 total seats - 12 already-purchased seats = 4).
5. The remainder (4) is the number of seats that is deducted from the pool of seats.

The customer purchased 13 seats for the **Reports** Feature in the update. However, the **Save** Feature has the highest accumulated number of seats. Therefore, only the **Save** Feature is considered when Sentinel LDK calculates the number of seats to deduct from the pool of seats.

**NOTE** A Feature with unlimited seats is regarded as having the value defined for the **Unlimited Concurrency** licensing parameter as described below.

## Unlimited Concurrency

Your Sentinel LDK Master license contains a licensing parameter called **Unlimited Concurrency** (also referred to as **Value of Unlimited Seats**). When you specify the concurrency value for a license as “unlimited” (for example, to create a “site” license), Sentinel LDK deducts the number of seats specified for licensing parameter from the HL pool of seats or SL pool of seats. This is typically 100 seats.

Given the following scenario:

- > A customer purchases 75 network seats for a Feature in a Product.
- > Later, the customer purchases unlimited network seats for that Feature.
- > The **Unlimited Concurrency** licensing parameter is set to 100 network seats.

Sentinel LDK charges this as an addition of 25 network seats. The pool of seats is decremented accordingly.

**NOTE** If you set or increase the number of network seats to a value greater than the **Unlimited Concurrency** value, the network seats pool will be decremented according to the value you specify. This charge may be greater than the value set for **Unlimited Concurrency**.

## Additional Information

- > When you purchase seats, Thales adds an extra 10% to the number of seats provided. This is intended to compensate for situations in which you reduce the number of seats at a customer site, or cancel a license on a computer on which Sentinel License Manager is located in order to activate on a different computer.
- > Once network seats have been consumed from the pool of seats, there is no action that will return the seats to the pool. Reducing the number of seats in a Product license or formatting an HL key that contains network seats does not return the seats to the pool. However, in certain circumstances, network seats that were erased will be restored to a license or HL key without being consumed again from the pool of seats:
  - If the number of seats in a Product license is reduced but then later increased (to the same number as before or higher), the earlier reduction is taken into consideration. Only seats (if any) that are beyond the original number will be consumed from the pool of seats.
  - If an HL key with network seats is formatted, the seats are lost. However, the original seats are restored without charge if the same Feature that owned the seats is later re-written to the HL key. Only additional seats (if any) beyond the original number will be consumed from the pool of seats.
- > The activation of a new license whose terms include concurrency will decrement both the New SL Key pool and the SL pool of seats.
- > If the terms of a new license include more seats than exist in your pool of seats, your customer will not be able to activate the license (if seats are required by your plan).

- > In a Modification Product, if you add concurrency to the license terms by specifying concurrency as 0, one network seat will become available for consumption in the following circumstances:
- The original license did not contain this Product, or
  - The original Product on which the Modification Product is based did not contain any Features (for example, a memory-only Product), or
  - The original Feature license terms were defined for local use only (no concurrency).

Depending on the type of key, this network seat will be consumed from either the **SL Pool of Seats** or **HL Pool of Seats** in the Sentinel LDK Master license and will be added to the Sentinel SL key or Sentinel HL (Driverless configuration) key. (Not relevant for Net or NetTime keys)

In all other cases, specifying 0 does not affect the number of seats consumed and does not change the license in the Sentinel SL or HL key.

## Unlocked Trialware Module

An unlocked trialware Product is a Product with an unlocked license that can be used for a relatively short period before the license expires. A protected application with an unlocked trialware license can be installed and operated on any number of computers. To continue using the application after the license expires, the user must purchase a production license. You can define the unlocked trialware Product so that the Product can be used either up to an absolute expiration date or for the defined number of days starting from the date of first use.

To define unlocked trialware Products with a duration of up to 90 days or up to 30 executions, you must purchase or subscribe to the Unlocked Trialware module for your Sentinel LDK Master license. (For extended durations, see the Unlocked Unlimited module, described below.)

**NOTE** The use of the Execution Count license type for Unlocked Products is only supported when working with Sentinel License Generation API.

The ability to create and distribute trialware Products is included in the Sentinel LDK – Demo and Starter packs. Vendors who want to experiment with Sentinel LDK can learn first-hand about unlocked trialware Products.

The maximum duration or maximum number of executions that you can define for any Feature in an Unlocked Product depends on the modules you have purchased or subscribed to for your Sentinel LDK Master license. For more information, see ["Defining Unlocked Products" on page 115](#).

## Unlocked Unlimited Module

The Unlocked license is for vendors who want to use Sentinel LDK to protect their applications against reverse engineering (by using Sentinel LDK Envelope) but do not require a traditional locked license.

An Unlocked license is similar to an unlocked trialware license. A protected application with an Unlocked license can be installed and operated on any number of computers. However, an Unlocked license can grant a perpetual license or a license for any length of time or any number of executions with no restrictions.

To generate Unlocked licenses, you must purchase or subscribe to the Unlocked Unlimited module for your Sentinel LDK Master license.

**NOTE** If you purchase or subscribe to the Unlocked Unlimited module, you can also create and distribute unlocked trialware Products without the need to purchase or subscribe to the Unlocked Trialware module.

The maximum duration or maximum number of executions that you can define for any Feature in an Unlocked Product depends on the modules you have purchased or subscribed to for your Sentinel LDK Master license. For more information, see ["Defining Unlocked Products" on page 115](#).

---

## V-Clock Module

V-Clock is a virtual clock that is available in Sentinel SL keys and in all Sentinel HL (Driverless configuration) keys except for Sentinel HL Basic keys. V-Clock is for vendors who want to use time-based licenses to protect their applications but do not want to provide a Sentinel HL Time key or Sentinel HL NetTime key. (These keys contain a real-time clock.)

The use of V-Clock with Sentinel SL keys and most Sentinel HL keys does not require a special license. However, to generate time-based licenses that depend on V-Clock in Sentinel HL Pro keys, you must purchase the V-Clock module for your Sentinel LDK Master license.

For more information on V-Clock, see ["How Sentinel LDK Protects Time-based Licenses With V-Clock" on page 346](#).

---

## AppOnChip Module

AppOnChip functionality provides significant protection for applications by moving code fragments from the application code to a Sentinel HL (Driverless configuration) key. This creates a strong binding between the protected application and the presence of the protection key, and limits reverse engineering of the protected code to black box analysis only.

The AppOnChip module is not required for applications that are licensed using Sentinel HL Max, Time, NetTime, Net, and Drive keys. For applications that are licensed using Sentinel HL Basic keys or Sentinel HL Pro keys, an annual or perpetual AppOnChip module must be obtained from Thales.

The AppOnChip module is typically placed on your Developer key. (For details, see ["Updating the Master License" on page 307](#).) The key that contains the AppOnChip license must be connected to the machine where Sentinel LDK Envelope or Sentinel Data Protection utility runs.

## Channel Partner Module

---

Basic functionality (associating an entitlement with a channel partner) does not require any special Sentinel LDK Master license modules. However, to use advanced channel partner functionality, you must obtain the Channel Partner module for your Sentinel LDK Master license.

The Channel Partner module is required to perform the following tasks:

- > Designate a user of Sentinel LDK-EMS as a Channel Partner user and associate the user with a specific channel partner.
- > Log in to Sentinel LDK-EMS by a Channel Partner user.
- > Associate a channel partner with a Product.

**NOTE** Before obtaining the Channel Partner module, you must prepare the Sentinel LDK-EMS database and resolve any conflicts caused by customers that are associated with multiple channel partners. For more information, see "Modifying the Sentinel LDK-EMS Database for Advanced Channel Partner Functionality" in the [Sentinel LDK Installation Guide](#).

## Reporting Module

---

The Reporting facility provides software vendors with the ability to produce real-time reports with valuable business information. The Custom Reports facility enables vendors to design their own reports to extract valuable information from the Sentinel LDK-EMS database.

Using the Custom Reports feature, managers can design reports to obtain data for analyzing how their software is used, the purchasing preferences of their customers, and information for profiling prospects and existing customers. The information can also be leveraged to maximize revenues from license renewals and to turn trial users into buyers.

The Reporting facility includes both predefined reports and the Custom Reports facility. Use of predefined reports does not require a specific license. However, use of the Custom Reports facility requires the Reporting module. This module is typically issued for a specific amount of time.

The ability to define, generate and view custom reports is included in the Sentinel License Development Kit – Demo and Starter. Vendors who are experimenting with Sentinel LDK can learn first-hand about the Custom Reports facility.

For information on the Reporting facility, see ["Generating Sentinel LDK Reports" on page 153](#).

## Cloud Licensing Module

---

Sentinel LDK provides an alternative for software-based licensing called *cloud licensing*. This model is based on the software-based protection keys, but provides an extension to the traditional model. Using the cloud licensing model, a protected application uses an identity string to log in to the SL license hosted on a vendor-managed or a

customer-managed license server machine.

For information on cloud licensing, see ["Cloud Licensing Using Sentinel Admin Control Center" on page 171](#).

To generate and update SL licenses that grant license access based on identity strings, your Sentinel LDK Master license must contain the Cloud Licensing module and the Product Activation module.

**NOTE** If cloud licensing is enabled in Sentinel LDK-EMS or Sentinel License Generation API, all license updates to SL keys will enable cloud licensing for the keys and will allow users (in the customer-hosted implementation) to freely manage their identities.

# APPENDIX B: Sentinel LDK Run-time Network Activity

This appendix describes the type of network activity that occurs in the communication between:

- > an application (protected using Sentinel LDK) and the local Sentinel License Manager (referred to as “local communications”).
- > the local Sentinel License Manager and one or more remote Sentinel License Managers (referred to as “remote communications”).

Details regarding local communications and remote communications are provided on the pages that follow.

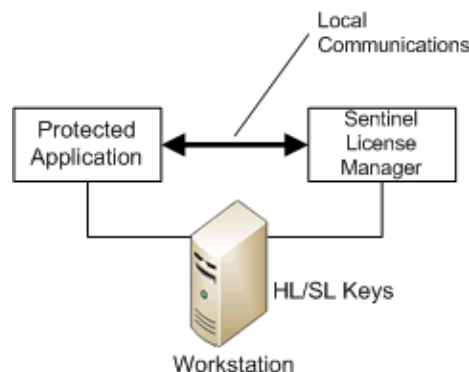
This section is intended to assist IT managers who want to understand how run-time activity on the network may impact the way they set up their network rules and policies.

Sentinel LDK communicates via TCP and UDP on socket 1947. This socket is IANA-registered exclusively for this purpose.

*In this appendix:*

- > ["Local Communications" below](#)
- > ["Remote Communications" on the next page](#)

## Local Communications



This section describes communication between a protected application and the local Sentinel License Manager service.

A protected application communicates only with Sentinel License Manager on the computer where the application is running, regardless of whether the Sentinel HL or SL Key is located on the same computer or on a remote computer.

**NOTE** Under Windows, Sentinel License Manager is a service that is launched automatically by **hasplms.exe**. Under Mac OS and Linux, the Sentinel License Manager is a process launched automatically by **hasplmd**.

Sentinel License Manager service opens socket 1947 for listening (both for UDP packets and TCP packets).

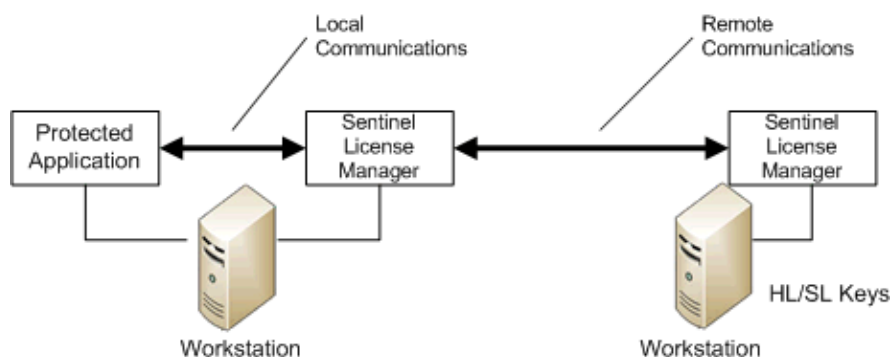
- > IPv4 sockets are always opened (Sentinel License Manager currently does not work without IPv4 installed).
- > IPv6 sockets are opened if IPv6 is available.

A protected application tries to connect to 127.0.0.1:1947 TCP to communicate with Sentinel License Manager. If an application uses multiple sessions, multiple concurrent TCP connections may exist. If a session is unused for a certain number of minutes (at least seven minutes, but the exact number depends on several factors), the session may be closed and automatically re-opened later in order to limit resources used by the application.

These local communications currently use IPv4 only.

The communication uses binary data blocks of varying size.

## Remote Communications



This section describes communication between the local Sentinel License Manager service and a remote Sentinel License Manager service.

This type of communication occurs when the protected application is running on a different computer from the computer where the Sentinel protection key is installed.

The protected application communicates only with the local Sentinel License Manager on the computer where the application is running, as described in ["Local Communications" on the previous page](#). The local Sentinel License Manager discovers and communicates with the License Manager on the computer containing the Sentinel protection key using one of the following methods:



- > If the option **Broadcast Search for Remote Licenses** is selected in the Admin Control Center (in the **Access From Remote Clients** tab of the Configuration page), the local Sentinel License Manager issues a UDP broadcast to local subnets on port 1947 using:
  - IPv4 (always)
  - IPv6 (if available)

The option **Broadcast Search for Remote Licenses** is selected by default.

- > For addresses specified in the Admin Control Center field **Remote License Search Parameters** or **Specify Search Parameters** (in the **Access From Remote Clients** tab of the Configuration page), the local License Manager does the following:
  - For a local **Admin License Manager**: The License Manager issues a UDP “ping” packet to port 1947 for all addresses specified. These addresses may be individual machine addresses or broadcast addresses.
  - For a local **Integrated License Manager** or **External License Manager**: The License Manager sends a TCP request to all individual addresses. If the field contains a broadcast address (xxx.xxx.xxx.255), the License Manager send a UDP broadcast to discover a running server at that broadcast address.

All Sentinel License Managers found by the discovery process are then connected via TCP port 1947, using IPv4 or IPv6 as detected during discovery, and data regarding the remote Sentinel protection keys are transferred.

This discovery process is repeated at certain intervals. (The interval size depending on a number of factors, but it is generally not less than five minutes.)

UDP packets sent and received in the discovery process contain the Sentinel License Manager GUID (40 bytes of payload data).

When starting or stopping a Sentinel License Manager, and when adding or removing a Sentinel protection key, a UDP notification packet is sent, containing the Sentinel License Manager GUID and a description of the changes encountered. This is done to allow other Sentinel License Managers to update their data before the next scheduled discovery process.

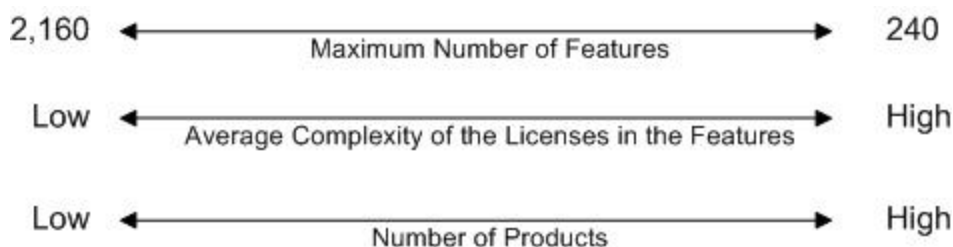
TCP packets between two Sentinel License Managers on different computers use HTTP with base-64 encoded data in the body section.

## APPENDIX C: Maximum Number of Features in a Sentinel HL Key

While the number of Features that can be written to a Sentinel SL key is unlimited, each Sentinel HL key can contain a certain maximum number of Features, depending on:

- > the type of HL key
- > the number of Products among which the Features are distributed
- > the complexity of the license type defined in each Feature:
  - Lowest complexity: Perpetual HL
  - Medium complexity: Perpetual HL + SL, Expiration Date
  - Highest Complexity: Execution Count, Time Period

For example, a Sentinel HL Max (Driverless configuration) key can contain:



This diagram illustrates that:

- > As you increase the number of higher-complexity license types on the key, the maximum number of Features that the key can contain decreases.
- > As you increase the number of Products on the key, the maximum number of Features that the key can contain decreases.

For information on the range of Features that each Sentinel HL key can contain, see the [Sentinel HL Data Sheet](#).

**NOTE** In Sentinel HL (Driverless configuration) keys, Features are stored in dynamic memory space. This space is shared between application data (the space available to you for your applications) and Features. All space that is not utilized for Features can be used for application data. For more information, see ["Defining Protection Key Memory Data" on page 111](#).

# APPENDIX D: How Sentinel LDK Detects Machine Cloning

This appendix describes the techniques employed by Sentinel LDK to prevent unauthorized use of protected software when the physical or virtual machine on which the software is installed is cloned. In addition, the appendix describes how you can examine the fingerprints from a machine to determine whether the machine and its protection keys were intentionally cloned.

This topic is only relevant for software protected with a Sentinel SL key. Software that is protected by a Sentinel HL key is not vulnerable to machine cloning.

For more information on protecting software against cloning, see ["Specifying the License Terms for Features in a Product" on page 110](#).

*In this appendix:*

- > ["How Clone Protection Schemes Work" below](#)
- > ["Using the "Platform Default" Scheme" on the next page](#)
- > ["Summary of Clone Protection Schemes" on page 327](#)
- > ["Requirements for Each Clone Protection Scheme" on page 328](#)
- > ["Clone Detection for Physical Machines" on page 330](#)
- > ["Clone Detection for Virtual Machines" on page 334](#)
- > ["How to Analyze a Clone Report" on page 340](#)
- > ["How to Clear the "Cloned" Status for a Product License" on page 345](#)

## How Clone Protection Schemes Work

One of the methods sometimes employed to enable the illegitimate use of licensed software is machine cloning. Machine cloning involves copying the entire image of one machine (including your software and its legitimate license) and duplicating it to one or more other machines. If there is no way to detect that the new image is running on different hardware than that on which it was originally installed, multiple instances of the software are available even though only a single license was purchased.

As part of the Activation process for a licensed Product, the Sentinel LDK License Manager creates a “fingerprint” of the computer on which the protected software is installed. This fingerprint contains hash values of a number of characteristics of the computer. This fingerprint (referred to as the *reference fingerprint*) is stored within the secure storage on the computer and is also returned to the Vendor in the C2V file. At the Vendor site, the fingerprint is stored as part of the license information in the Sentinel LDK-EMS database.

Each time the end user starts the protected software, the Sentinel LDK License Manager creates a new fingerprint of the computer (referred to as the *system fingerprint*) and compares it to the reference fingerprint.

If the system and reference fingerprints are identical or sufficiently close (as described in this appendix), Sentinel LDK allows the protected software to operate.

When clone detection is enabled for a Product in Sentinel LDK, the License Manager checks for cloning using the criteria described in this appendix. If cloning is detected, Sentinel LDK disables the license. As a result, the end user is unable to operate the software for which a cloned license has been detected.

## Using the "Platform Default" Scheme

---

A clone protection scheme defines which factors are considered by the Sentinel License Manager to determine whether a given Sentinel SL key has been cloned.

Sentinel LDK offers several different clone protection schemes to protect applications that execute on physical machines and on virtual machines. These schemes are designed to accommodate a variety of circumstances. For example, schemes are available for applications that run on PCs or on Microsoft Azure virtualization platforms. New schemes are added periodically as environments are added and evolve. (For advanced users, the clone protection schemes are described in detail in this appendix.)

Sentinel LDK-EMS and Sentinel LDK License Generation API both allow you to select a scheme called **Platform Default** instead of selecting a specific clone protection scheme for a Product. For more information, see ["Simplified Clone Protection" on page 115](#).

When you select **Platform Default** as the scheme for a virtual machine or a physical machine (or both), Sentinel LDK automatically applies the most appropriate clone protection scheme for each end user based on the following parameters:

- > the environment in which the protected application is installed.
- > the earliest version number in use by the vendor's customer base from among the following:
  - For SL AdminMode keys: the earliest version number of the Run-time Environment (RTE) in use by the vendor's customer base.
  - For SL UserMode keys: the earliest version of Sentinel LDK used to provide the External or Integrated License Manager (LM) in use by the vendor's customer base.

**NOTE** If you plan to ensure that each customer receives the latest RTE or LM when you deliver a Product license, you can assume that the earliest version number for your customer base is the current version.

The environment is determined automatically by Sentinel LDK. However, it is the responsibility of the vendor to specify the appropriate version number for the customer base.

Specifying a later version number results in the selection of a more advanced clone protection scheme. However, the selected version number must not be higher than the version numbers that exist where the Product license will be installed.

In Sentinel LDK-EMS, you specify the version number in the configuration parameter **Minimum RTE/API Version**. In Sentinel License Generation API, the version is specified as part of the `<minimum_rte_api_version>` tag in the Product definition. For more information, see ["Minimum RTE/API Version" on page 201](#).

**NOTE** The **Platform Default** clone protection scheme is applied automatically when a license is detached from the network. This occurs regardless of whether clone protection was enabled for the Product license on the license server from which the license was detached.

Use the table below to determine which value you should specify in Sentinel LDK-EMS or Sentinel License Generation API, as follows:

1. Determine which column in the table is consistent with the earliest version number of the RTE and LM in use for your customer base as described above.
2. From the column that you selected, use the specified value in Sentinel LDK-EMS or in Sentinel License Generation API.

For example:

- > If the earliest version of the RTE in use by your customer base is 7.70, but the earliest LM for SL UserMode keys is from Sentinel LDK 7.5, use the value **7.50** for **Minimum RTE/API Version** or for the `<minimum_rte_api_version>` tag.
- > If you plan to upgrade your customers to RTE version 7.100 or to the LM from Sentinel LDK 7.10 with each new license or license update, use the value **7.100**.

The following table indicates which clone protection scheme Sentinel LDK selects for each value:

Operating System and Environment	For RTE 6.40–7.4x or Sentinel LDK 6.4–7.4x	For RTE 7.5x or Sentinel LDK 7.5x	For RTE 7.60-7.9x or Sentinel LDK 7.6-7.9x	For RTE 7.10x and later or Sentinel LDK 7.10x	For RTE 8.11x and later or Sentinel LDK 8.0x and later
	Value for "Minimum RTE/API Version" for Sentinel LDK-EMS or License Generation API:				
	6.40	7.50	7.60	7.100	8.11
Windows, Linux, Mac (Excluding Docker Containers)	PM: PMType1 VM: VMType1	PM: PMType2** VM (SL AM): VMType2** VM (SL UM): VMType1**			
Windows, Linux (Docker Containers)	VM: VMType1	VM (SL AM): VMType2 VM (SL UM): VMType1		VM: VMType4*	
Linux (LXC Container)	Not relevant			VM: VMType4	
Android	PM: PMType3 VM: No Clone Protection		PM: PMType4 VM: No Clone Protection		

**NOTE** For licensing on a hosted virtual machine (Amazon, Google, Azure, or Alibaba Cloud), the **VMType3** scheme is applied in the following situation:

- > You select **Platform Default** as the clone protection scheme for virtual machines.
- > In Sentinel LDK-EMS or Sentinel License Generation API, you choose **8.11** for the **Minimum RTE/API Version**.

This applies regardless of whether or not the license is in a Docker container on one of the hosted virtual machines described above. In any other circumstances for Docker, the **VMType4** scheme is applied for **Platform Default**.

#### Table Legend:

PM - physical machine

VM - virtual machine

SL AM - SL AdminMode key

SL UM - SL UserMode key

\* **VMType4** is used if the license is activated inside the Docker container. If the license is activated by the host machine, the clone protection scheme is one of the other types, depending on the host machine.

\*\* These clone protection schemes will also be used if the minimum RTE/API version is not specified.

## Summary of Clone Protection Schemes

Several schemes exist in Sentinel LDK to create fingerprints for physical and virtual machines. These schemes provide different levels of protection to satisfy the various sets of requirements that may exist in your organization.

The lists below summarize the various clone protection schemes available. A more detailed description of each clone protection scheme is provided later in this appendix.

### Summary of Schemes for Physical Machines

- > **Platform Default:** Instructs Sentinel LDK to automatically apply the most appropriate clone protection scheme for each end user based on various parameters. For details, see ["Using the \"Platform Default\" Scheme\" on page 324](#).
- > **PMType1:** Uses two components to verify fingerprints: hard drive serial number and motherboard ID. For details, see [\"PMType1 Scheme\" on page 330](#).
- > **PMType2:** Uses various components such as CPU, ethernet card, optical drive, and PCI card slot peripherals, along with the hard drive serial number and motherboard ID to verify fingerprints. This scheme provides enhanced reliability against false positive clone detection and maintains the inherent security of the scheme. For details, see [\"PMType2 Scheme\" on page 330](#).
- > **PMType3:** For Android applications. Uses three components to verify fingerprints: CPU model, CPU serial number, and internal storage serial number. For details, see [\"PMType3 Scheme\" on page 331](#).
- > **PMType4:** For Android applications. Uses up to five components to verify fingerprints. Additionally, this scheme allows the end user to uninstall and reinstall the protected application in many instances without vendor assistance. For details, see [\"PMType4 Scheme\" on page 332](#).
- > **FQDN:** Uses only the machine's FQDN (Fully Qualified Domain Name) to verify fingerprints. For details, see [\"FQDN Scheme\" on page 333](#).

**NOTE** On MAC machines, FQDN licenses are bound to LocalHostName, and the value of LocalHostName should not be empty.

- > **Custom:** You can define your own clone protection scheme that includes criteria that you select from a list. You also specify the minimum number of the selected criteria that must match when validating the license. For details, see ["Custom Scheme" on page 333](#).

## Summary of Schemes for Virtual Machines

- > **Platform Default:** Instructs Sentinel LDK to automatically apply the most appropriate clone protection scheme for each end user based on various parameters. For details, see ["Using the "Platform Default" Scheme" on page 324](#).
- > **VMType1:** Uses three components to verify fingerprints: Virtual MAC address, CPU characteristics, and UUID. For details, see ["VMType1 Scheme" on page 334](#).
- > **VMType2:** Uses four components to verify fingerprints: Virtual MAC address, CPU characteristics, UUID, and Snapshot Rollback Detection. This scheme has additional restrictions that are described in ["Clone Detection for Virtual Machines" on page 334](#). This scheme prevents attacks (again a protected application) that are based on virtual machine rollback snapshots. For details, see ["VMType2 Scheme" on page 336](#).
- > **VMType3:** Provides strong and reliable clone protection for cloud computing services such as Amazon EC2 and the Microsoft Azure virtualization platform. For details, see ["VMType3 Scheme" on page 337](#).
- > **VMType4:** Provides strong and reliable clone protection for Docker containers. For details, see ["VMType4 Scheme" on page 338](#).
- > **FQDN:** Uses the machine's FQDN (Fully Qualified Domain Name) to verify fingerprints. This scheme provides increased reliability and provides flexibility of operation in a server virtualization environment. For details, see ["FQDN Scheme" on page 339](#).
- > **Custom:** You can define your own clone protection scheme that includes criteria that you select from a list. You also specify the minimum number of the selected criteria that must match when validating the license. For details, see ["Custom Scheme" on page 339](#).

**NOTE** The clone protection provided by the **VMType1** and **FQDN** protection schemes are based on the following assumption: The customer's IT department follows best practices to avoid the collisions that would result from cloned machines that have identical UUID, MAC addresses or hostnames.

If you are concerned that your customers may be willing to accept collisions in order to attempt to bypass clone protection, consider one of the other Sentinel LDK solutions that provides a different tradeoff of security and convenience and is not affected by such deployment. A remote license (SL AdminMode or Sentinel HL) will provide the higher level of security that you require.

## Requirements for Each Clone Protection Scheme

Each clone protection scheme requires one of the following as described in the table below:



- > **SL AdminMode keys and SL Legacy keys:** A minimum version of the Sentinel Run-time Environment.
- > **SL UserMode keys:** A minimum version of the External License Manager or Integrated License Manager from Sentinel LDK.
- > **Android applications:** A minimum version of the Integrated License Manager from Sentinel LDK.

Clone Protection Scheme	Integrated/External LM From Sentinel LDK Version	Integrated LM From Sentinel LDK Version	Minimum Required Version of Sentinel Run-time Environment		
			Windows	Linux	Mac
PMTYPE1	7.1	na	6.61	2.4.1	7.1
PMTYPE2	7.1	na	6.61	2.4.1	7.1
PMTYPE3	7.3	7.3	Not applicable		
PMTYPE4	7.6	7.6	Not applicable		
FQDN	7.1	Not applicable	6.61	2.4.1	7.1
VMType1	7.1	Not applicable	6.61	2.4.1	7.1
VMType2	7.5	Not applicable	7.51	7.51	7.51
VMType3	7.7	Not applicable	7.61 (Azure platform)		Not applicable
VMType4	7.10	Not applicable	7.100	7.100	7.100
Custom	7.9	7.9	7.90	7.90	7.90

# Clone Detection for Physical Machines

This section provides a detailed description of the clone protection schemes that are available to protect against the cloning of physical machines.

## Platform Default Scheme

The **Platform Default** scheme instructs Sentinel LDK to automatically apply the most appropriate clone protection scheme for each end user based on various parameters. For details, see ["Using the "Platform Default" Scheme" on page 324](#).

## PMType1 Scheme

The **PMType1** scheme uses two components to verify fingerprints: hard drive serial number and motherboard ID.

If *either* the hard drive serial number *or* the motherboard ID does not match the characteristics in the fingerprint in the secure storage, Sentinel LDK License Manager still allows the protected software to operate. Sentinel LDK recognizes that situations occur where an end user has a legitimate reason for replacing one of these components in the user's computer. This policy possibly enables a user to operate protected software on a cloned computer. However, this policy also frees the Vendor from dealing with numerous support calls from users who have replaced a component in their computer. Such calls would otherwise generate costly support cases for the Vendor's customer support organization.

If *both* the hard drive serial number *and* the motherboard ID do not match the characteristics in the fingerprint of the license, Sentinel LDK regards computer as a clone and prevents the protected software from operating. (See the table that follows.)

Characteristics Compared	Comparison Results			
Hard drive serial number	Identical	Different	Identical	Different
Motherboard ID	Identical	Identical	Different	Different
Sentinel LDK Behavior: The software is...	Launched	Launched	Launched	Disabled

**Supported operating systems:** Windows, Linux, and Macintosh

## PMType2 Scheme

The **PMType2** scheme uses various components such as CPU, ethernet card, optical drive, PCI card slot peripherals (for example: display, storage, network, multimedia) along with the hard drive serial number and motherboard ID to verify fingerprints on a physical machine.

Each component that makes up the reference fingerprint is assigned a weighted value. Sentinel LDK performs the following computations:

- > A = total for the weighted values of all components in the reference fingerprint.
- > B = total for the weighted values of all components in the system fingerprint that match components in the reference fingerprint.
- > matching percentage =  $(B/A) * 100$

Sentinel LDK computes a *required percentage* based on the level of agreement that is found between the hard drive serial number and motherboard ID in the reference fingerprint and in the system fingerprint.

If the matching percentage reaches the required percentage, the protected application is allowed to execute.

**NOTE** Thales recommends the use of **PMType2** over **PMType1** because **PMType2** is a more advanced scheme that provides better reliability and security.

## PMType3 Scheme

The **PMType3** scheme is specifically for Android applications.

The requirements of the **PMType3** scheme are:

- > The internal storage serial number must match the characteristics in the fingerprint in secure storage.
- > If the internal storage serial number is absent, the CPU information must match the characteristic in the fingerprint in secure storage.

If the protected application is re-installed on the user's device, the user must send a C2V file to the vendor and receive a new V2C file in return in order to activate the product license.

**NOTE** **PMType3** is a legacy clone protection scheme. Thales recommends that you use **PMType4** instead. **PMType4** is not disruptive if app re-installation is required, and it provides the same level of security and reliability as **PMType3**.

The table that follows describes the requirements of the **PMType3** scheme in detail.

Characteristics Compared	Comparison Results			
Internal Storage serial number	Identical	Different	Absent	Absent
CPU information	Not relevant	Not relevant	Identical	Different
Sentinel LDK Behavior: The software is...	Launched	Disabled	Launched	Disabled

## PMType4 Scheme

The **PMType4** scheme is a more advanced scheme for Android applications. This scheme uses the Android ID as the primary factor in checking for clones. When available, the internal storage serial number, the Android serial number and Android first boot time are also used.

(For Android 10 and later, due to platform restrictions, only the Android ID is available.)

This scheme allows for situations where the end user uninstalls and then reinstalls the protected application.

Typically, after a reinstall, the user is required to request a new V2C file from the vendor to re-enable the license for the application. This is required because some licenses may restrict the number of executions or may be time-restricted based on the installation date.

With the **PMType4** scheme, if the license is perpetual or is time-restricted based on an absolute expiration date, a new V2C file is not required. As a result, both the vendor and the customer are saved the effort of resolving licensing issues unnecessarily.

The table that follows demonstrates the requirements for the operation of an application that is protected using the **PMType4** scheme.

Characteristic s Compared	Comparison Results						
<b>Android ID</b>	Identical	Different	Absent	Absent	Absent	Absent	Absent
<b>Internal Storage serial number</b>	Not relevant	Not relevant	Identical	Different	Absent	Absent	Absent
<b>CPU information</b>	Identical	Not relevant	Not relevant	Not relevant	Identical	Identical	Different
<b>Android serial number</b>	Not relevant	Not relevant	Not relevant	Not relevant	Identical	Not relevant	Not relevant
<b>Android first boot time</b>	Not relevant	Not relevant	Not relevant	Not relevant	Not relevant	Identical	Not relevant
<b>Sentinel LDK Behavior: The software is...</b>	Launched	Disabled	Launched	Disabled	Launched	Launched	Disabled

## FQDN Scheme

The FQDN scheme uses only the machine's FQDN (Fully Qualified Domain Name) to verify fingerprints on a physical machine. If the FQDN in the reference fingerprint matches the FQDN in the system fingerprint, the protected Software is launched.

Use of the FQDN scheme reduces possible false-positive clone detection that may result from changes to hardware devices. Such changes may be flagged as the result of license cloning to another machine.

The security level of the FQDN scheme is limited, as domains and hostnames can be spoofed. FQDN is useful in scenarios where you trust the users to not attempt to bypass the licensing, but where license compliance is important.

Use of the FQDN scheme is preferred primarily in networks where a domain is used. This is typical in the networks of corporate users. On standalone machines, domains are typically not used; therefore, only the hostname is used for locking.

For higher security, use either of the following:

- > SL key with the **Platform Default** clone protection scheme. This does not require connectivity, but significant changes to the hardware may result in false-positive clone detection.
- > Cloud licensing. This scheme is resistant to hardware changes on the client machine but requires at least occasional connectivity.

## Custom Scheme

You can define a custom clone protection scheme that includes one or more criteria that you select from the table that follows.

Criteria	Notes
<b>CPU</b>	CPU information. CPU UID is excluded
<b>Ethernet address</b>	MAC address
<b>FQDN</b>	Fully Qualified Domain Name. Not supported for Android
<b>Hard disk</b>	Hard disk ID (on a PC) or SD card ID (Android device)
<b>IP address</b>	IP address
<b>Machine ID</b>	Motherboard (on a PC) or Android serial number (or Android first boot if serial number is not available)
<b>Security Identifier (SID)</b>	Microsoft Windows Security Identifier (Windows machine only)

You also specify how many of the selected criteria must match when the License Manager validates the license. For example, you can select six criteria from the table, but specify that only three of the six must match in order to validate the license.

You can define custom schemes using either Sentinel LDK-EMS or Sentinel License Generation API.

In Sentinel LDK-EMS, you assign a name for each custom scheme. This simplifies the process of reusing the custom scheme for additional Products.

## Clone Detection for Virtual Machines

---

This section provides a detailed description of the clone protection schemes that are available to protect against the cloning of virtual machines.

### Platform Default Scheme

The **Platform Default** scheme instructs Sentinel LDK to automatically apply the most appropriate clone protection scheme for each end user based on various parameters. For details, see ["Using the \"Platform Default\" Scheme\" on page 324](#).

### VMType1 Scheme

Clone detection for software installed on a virtual machine must employ a different technique than that used for physical machines.

The two most important fingerprint characteristics - the physical hard drive serial number and the physical motherboard ID - are not accessible to software running on the virtual machine. Instead, the virtual machine has a virtual hard drive and a virtual motherboard.

On a cloned virtual machine, the characteristics of these virtual components are identical to the source virtual machine. As a result, these characteristics are not suitable for use when creating the fingerprint at the time the protected software is activated or subsequently operated.

The **VMType1** scheme relies on three different parameters for verifying fingerprints on a virtual machine: the virtual MAC address, CPU characteristics, and UUID of the virtual image. Each of these parameters is discussed below.

### Virtual MAC Address

Each physical network adapter or network card has a unique identifier, but this identifier is not accessible to a virtual machine running on the computer. Instead, each virtual machine is assigned a unique virtual MAC address.

Within a network, each virtual machine must possess a unique MAC address. If a user clones a virtual machine and installs it on a second computer within the same network, working on either the original or the cloned virtual machine will be impractical as the two machines will constantly cause network collisions.

## CPU Characteristics

In **desktop/workstation environments** such as VMware workstation or VMware player, the desktop virtualization software does not expose the ability to virtualize the CPU. This increases the difficulty for a user to bypass the protection by attempting to create a virtual copy of the source computer. A number of CPU characteristics are available for inclusion in the virtual machine fingerprint, including: processor make, model and speed.

Due to the large number of different processors available in the market, the likelihood of two different desktop computers having completely identical CPU characteristics is low.

In **centrally managed virtual infrastructures** (also called *server based virtualization*), hardware clusters can be virtualized. In this environment, the virtual infrastructure does not always utilize a single, fixed set of physical hardware resources. Instead, it utilizes a shared pool of resources. For the most common types of clustered environments, where live migration capabilities are typically required, a requirement usually exists for different hosts in the cluster to have identical CPU characteristics. Solutions such as VMware vCenter Server provide the ability to enable CPU masking to improve compatibility for the high availability and fault tolerance virtualization features. CPU masking allows host machines with different CPU characteristics to be used in the cluster, while providing common (masked) CPU characteristics across all hosts in the cluster. Therefore the CPU characteristics do not change when virtual machine migrates across the hosts in a cluster. This enables licensed applications to continue working when migrated from one host to another within a cluster. However, this type of environment is restricted to a limited subset of CPU types. In addition, the migration can only be performed when the target computer contains physical CPU whose capabilities match or exceed the characteristics of the virtual CPU.

## UUID of the Virtual Machine

This is used as a means of unique identification of the virtual machine with the majority of virtual machines technologies. The UUID consists of a 16-byte (128-bit) number. Each virtual machine is assigned a different UUID.

When a user makes a clone of a virtual image or copies a virtual machine from one location to another, a new UUID value is generated for the new virtual image or virtual machine.

None of the three characteristics used by this scheme to create a virtual machine fingerprint is absolutely tamper-proof.

The protection against cloning provided by Sentinel LDK for virtual machines is not as secure as the protection provided for physical machines. You have the option of blocking the protected software from running on most popular virtual machines by clearing the **Virtual Machine** check box in the Define License Terms dialog box in Sentinel LDK-EMS.

However, when checking the fingerprint for cloning, Sentinel LDK examines all of these characteristics. If *one* (or more) of these characteristics does not match the characteristics in the fingerprint of the license, Sentinel LDK prevents the protected software from operating. Thus, the combination of these parameters in the fingerprint provides protection against cloning. (See the table that follows.)

		Comparison Results			
Characteristics Compared	Virtual MAC Address	Identical	Different	Not relevant	Not relevant
	CPU Characteristics	Identical	Not relevant	Different	
	UUID	Identical		Not relevant	Different
Sentinel LDK Behavior: The software is...		launched	disabled	disabled	disabled

In a typical business environment (where computers in a given location are on the same network), the requirement for a unique virtual MAC address make cloning impractical.

For server virtualization, or virtualized cluster where the cluster is typically managed by the virtualized management solution (such as VMware vCenter), UUID acts as additional deterrent to running a cloned virtual image.

For computers on different networks or computers that are not networked, the likelihood of a cloned virtual machine sharing identical CPU characteristics with the original virtual machine is low.

The method employed by this scheme to protect against cloning of virtual machines is effective for all types of virtual machine software commonly used by organizations.

## VMType2 Scheme

This scheme provides the same protection that is provided by **VMType1**. In addition, this scheme prevents attacks (against a protected application) that are based on virtual machine rollback snapshots. The scheme enables the protected application on a virtual machine to detect that a time shift event may have occurred.

The table that follows describes the circumstances under which the protected application is disabled with the **VMType2** scheme.



		Comparison Results				
<b>Characteristics Compared</b>	<b>Virtual MAC Address</b>	Identical	Different	Not relevant	Not relevant	Not relevant
	<b>CPU Characteristics</b>	Identical	Not relevant	Different		
	<b>UUID</b>	Identical		Not relevant	Different	
	<b>Rollback Snapshot Detected</b>	No			Not relevant	Yes
<b>Sentinel LDK Behavior: The software is...</b>		launched	disabled			

The VMType2 scheme is only supported under the following circumstances:

- > Run-time Environment v.7.5 or later is present on the virtual machine where the protected application is running.
- > The locking type is SL AdminMode.

The scheme is supported on Windows 8, Windows 10, Windows Server 2012 R2, and later versions of Windows Server, with the supported versions of the following virtual machines:

- > VMware Player, Workstation, and ESXi
- > Hyper-V Server

In addition, the scheme is supported on certain earlier versions of Windows with Hyper-V Server if Hyper-V integration services from Windows 8 or Windows Server 2012 is installed.

For more information, see: [https://msdn.microsoft.com/en-us/library/jj643357\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/jj643357(v=vs.85).aspx)

For other virtual machine clients that do not support **VMType2**, this scheme will be handled as if you had selected the **VMType1** scheme.

## VMType3 Scheme

The **VMType3** clone protection scheme provides strong and reliable clone protection for cloud computing services such as Amazon EC2, Microsoft Azure, Google Cloud Platform and Ali Cloud. This scheme addresses the following situations:

- > The scheme ensures that a protected application in a cloud computing service cannot be used if the license is copied from one virtual machine to another.
- > The scheme ensures that an SL UserMode licenses is protected against misuse by UserMode secure storage wipeout.

Prerequisites for use of the **VMType3** clone protection scheme:

- > The machine for which the license is generated must be a cloud computing service.
- > The minimum required version of the Run-time Environment on the target machine is 7.61. (For Amazon EC2, the minimum required version is 8.13.)
- > The latest fingerprint of the target machine should contain items `mainboard_uid2` and `vm_info2`. This fingerprint should be obtained after installation of the required version of the Run-time Environment.
- > The License Generation library version must be 7.61 or later in order to generate a license with the **VMType3** clone protection scheme. (For Amazon EC2, the minimum required version is 8.0.45378.60000.)

		Comparison Results		
<b>Characteristics Compared</b>	<b>mainboard_uid2</b>	Different	Not relevant	Identical
	<b>secure_storage_uid</b>	Not relevant	Different	Identical
<b>Sentinel LDK Behavior: The protected application is...</b>		disabled	disabled (Secure Storage wipeout most likely occurred)	launched

**NOTE** Start / Stop / Restart from Azure infra will not be reported as **Cloned**.

**NOTE** For licensing on a hosted virtual machine (Amazon, Google, Azure, or Alibaba Cloud), the **VMType3** scheme is applied in the following situation:

- > You select **Platform Default** as the clone protection scheme for virtual machines.
- > In Sentinel LDK-EMS or Sentinel License Generation API, you choose **8.11** for the **Minimum RTE/API Version**.

This applies regardless of whether or not the license is in a Docker container on one of the hosted virtual machines described above. In any other circumstances for Docker, the **VMType4** scheme is applied for **Platform Default**.

## VMType4 Scheme

This scheme is intended primarily for Docker containers, but it is compatible with other virtual machines.

When the **VMType4** scheme is selected, the characteristics checked are:

- > Virtual MAC address
- > CPU characteristics
- > UUID

- > Hard drive serial number

The values for these characteristics must match in the reference fingerprint and the system fingerprint. If there is any mismatch, the protected application is disabled.

## FQDN Scheme

The **FQDN** scheme uses only the machine's FQDN (fully qualified domain name) to verify fingerprints on a virtual machine. If the FQDN in the reference fingerprint matches the FQDN in the system fingerprint, the protected Software is launched.

The **FQDN** clone protection scheme provides a solution for virtual machine live migration. It allows the guest virtual machine to freely migrate between different physical hosts, while allowing accurate license enforcement to continue. Virtual machine live migration does not cause false-positive clone detection.

The security level of the FQDN scheme is limited, as domains and hostnames can be spoofed. FQDN is useful in scenarios where you trust that the user will not attempt to bypass the licensing, but where license compliance is important.

Use of the FQDN scheme is preferred primarily in networks where a domain is used. This is typical in the networks of corporate users. On standalone machines, domains are typically not used; therefore, only the hostname is used for locking.

For higher security, use either of the following:

- > SL key with the **Platform Default** clone protection scheme. This does not require connectivity, but significant changes to the hardware may result in false-positive clone detection.
- > Cloud licensing. This scheme is resistant to hardware changes on the client machine but requires at least occasional connectivity.

### NOTE

- > For cloud licensing on a hosted virtual machine (Amazon, Google or Azure), Thales recommends that you use the **VMType3** scheme. For cloud licensing on other hosted virtual machines, Thales recommends that you use a custom scheme based on machine ID.
- > The **FQDN** scheme should never be selected for Docker containers if the License Manager service executes inside the container. The fingerprint will contain a random value.

## Custom Scheme

You can define a custom clone protection scheme that includes one or more criteria that you select from the table that follows.

Criteria	Notes
CPU	CPU information
Ethernet address	MAC address
FQDN	Fully Qualified Domain Name
VM generation ID	Attribute of a Windows VM that helps to prevent misuse of a VM snapshot
IP address	IP address
Machine ID	Motherboard (on a PC) or Android serial number (or Android first boot if serial number is not available)
Security Identifier (SID)	Microsoft Windows Security Identifier (Windows machine only)

You also specify how many of the selected criteria must match when the License Manager validates the license. For example, you can select six criteria from the table, but specify that only three of the six must match in order to validate the license.

You can define custom schemes using either Sentinel LDK-EMS or Sentinel License Generation API.

In Sentinel LDK-EMS, you assign a name for each custom scheme. This simplifies the process of reusing the custom scheme for additional Products.

## How to Analyze a Clone Report

Under certain circumstances, the Sentinel License Manager may report that cloning was detected even if intentional cloning did not occur. If an end user complains to you that they were blocked from using a Product due to cloning, use the procedure described below to analyze information from the user's machine. This helps you determine whether the Product was blocked due to a false-positive clone report or whether an actual clone attempt occurred.

### To generate a clone report:

1. Instruct the end user to generate a C2V file for the relevant protection key using RUS (described in ["Sentinel Remote Update System \(RUS\)" on page 147](#)) or Admin Control Center and to send the file to you. Alternatively, you can include a function in your application to generate a C2V file by calling the `GetInfo` function in the Sentinel Licensing API.
2. In Sentinel LDK ToolBox or Sentinel License Generation API:
  - a. Call `sntl_ig_initialize` to initialize a session.
  - b. Call `sntl_ig_decode_current_state` to read the C2V file and decode the current state of the protection keys on the end user's machine.

### 3. Examine the decoded XML output of the C2V file as described below.

The decoded XML output of the C2V file contains information regarding all the protection keys on the machine. For each protection key, the XML output contains:

- > The fingerprint of the machine from the time the protection key was activated ("reference fingerprint")
- > The fingerprint of the machine from the time cloning was detected ("system fingerprint")
- > Information regarding the Product licenses in the key

If the Sentinel License Manager detects that a protection key was cloned, the XML output for the key contains code similar to this:

```
<clone_detected machine_type="Physical">Yes</clone_detected>
```

For each Product that is regarded as cloned, the section in the XML output for that Product contains this code:

```
<fingerprint_change>Cloned</fingerprint_change>
```

Each Product can use a different predefined or custom clone protection scheme for physical and for virtual machines. A Product can contain both a predefined and a custom clone protection scheme. (For example, a predefined scheme for physical machines and a custom scheme for virtual machines.)

The clone protection scheme indicates which factors are considered by the License Manager when it checks to determine whether a machine (together with its protection keys) was cloned. For each Product license in the XML output, determine which clone protection scheme is used, and then compare the relevant factors in the reference fingerprint and the system fingerprint. The results of the comparisons should enable you to determine whether an intentional attempt was made to clone a machine and its protection keys. Clone protection schemes are described earlier in this appendix.

You can determine whether a given machine in a fingerprint is a physical machine or a virtual machine. The following tag in a fingerprint indicates that the corresponding machine is a physical machine:

```
<criteria>
  <name>vm_info</name>
  <value>1294737779</value>
</criteria>
```

If the value indicated in the tag above is anything other than 1294737779, the fingerprint is from a virtual machine.

For more information on the decoded XML output file, see the [Sentinel License Generation API Reference](#).

## Example 1 (Predefined Clone Protection Schemes)

Suppose you receive a C2V file from a customer who complains that protected applications will not run.

You decode the C2V file using ToolBox and then examine the generated XML code in a text editor. The XML code in the file indicates that the license is cloned.

The clone protection schemes are PMType1 and VMType2 as seen below.

```
<clone_protection_ex>
  <physical_machine>PMType1</physical_machine>
  <virtual_machine>VMType2</virtual_machine>
</clone_protection_ex>
```

The system fingerprint and the reference fingerprint from the XML code are shown below side-by-side.

<system_fingerprint>	<reference_fingerprint>
<pre>&lt;raw_data&gt;Mm03m1f1B5BSCNgDBzS&lt;/raw_data&gt; &lt;fingerprint_info&gt;   &lt;criteria&gt;     &lt;name&gt;cpu&lt;/name&gt;     &lt;value&gt;1567025546&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;mainboard&lt;/name&gt;     &lt;value&gt;274685399&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;mainboard_uid&lt;/name&gt;     &lt;value&gt;3138942878&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;hdd&lt;/name&gt;     &lt;value&gt;3692089826&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;hdd_uid&lt;/name&gt;     &lt;value&gt;2413362815&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;ethernet_uid&lt;/name&gt;     &lt;value&gt;4087917752&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;vm_info&lt;/name&gt;     &lt;value&gt;1294737779&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;secure_storage_uid&lt;/name&gt;     &lt;value&gt;701117991&lt;/value&gt;   &lt;/criteria&gt; &lt;/fingerprint_info&gt;</pre>	<pre>&lt;fingerprint_control_type&gt;ISV Mar&lt;/fingerprint_control_type&gt; &lt;raw_data&gt;MXhJSWHQgXHZSpBSWAYcJfI&lt;/raw_data&gt; &lt;fingerprint_info&gt;   &lt;criteria&gt;     &lt;name&gt;cpu&lt;/name&gt;     &lt;value&gt;1846611331&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;mainboard&lt;/name&gt;     &lt;value&gt;129528607&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;mainboard_uid&lt;/name&gt;     &lt;value&gt;1410961151&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;hdd&lt;/name&gt;     &lt;value&gt;3692089925&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;hdd_uid&lt;/name&gt;     &lt;value&gt;2413362511&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;optical_drive&lt;/name&gt;     &lt;value&gt;521645421&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;ethernet_uid&lt;/name&gt;     &lt;value&gt;4219815536&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;vm_info&lt;/name&gt;     &lt;value&gt;1294737779&lt;/value&gt;   &lt;/criteria&gt; &lt;/fingerprint_info&gt;</pre>

As seen in the code:

- > The motherboard IDs (mainboard\_uid) in the system and reference fingerprints are not identical.
- > The hard drive serial numbers (hdd\_uid) in the system and reference fingerprints are not identical.
- > The value of vm\_info indicates that both fingerprints are from physical machines.

Therefore, these fingerprints satisfy the criteria for the **PMType1** scheme to indicate that the license is from a cloned machine.

## Example 2 (Custom Clone Protection Schemes)

Given that you receive a C2V file from a customer who complains that protected applications will not run.

You decode the C2V file using the ToolBox and then examine the generated XML code in a text editor. The XML code in the file indicates that the license is cloned.

The clone protection schemes are both custom as seen below.

```
<clone_protection_ex>
  <physical_machine_custom>
    <name>PM Custom</name>
    <criteria minimum="2">
      <name>ethernet_address</name>
      <name>sid</name>
    </criteria>
  </physical_machine_custom>
  <virtual_machine_custom>
    <name>VM Custom</name>
    <criteria minimum="2">
      <name>ethernet_address</name>
      <name>sid</name>
    </criteria>
  </virtual_machine_custom>
</clone_protection_ex>
```

For both physical and virtual machines, the criteria are ethernet address and security identifier (SID). Since the **minimum** attribute is set to 2, both criteria must match in order to validate the license.



The system fingerprint and the reference fingerprint from the XML code are shown below side-by-side.

<system_fingerprint>	<reference_fingerprint>
<pre> &lt;raw_data&gt;MXhJSUslaJDKFxBSYk2PS &lt;fingerprint_info&gt;   &lt;criteria&gt;     &lt;name&gt;cpu&lt;/name&gt;     &lt;value&gt;3920118366&lt;/value&gt;   &lt;/criteria&gt;   ...   &lt;criteria&gt;     &lt;name&gt;mainboard&lt;/name&gt;     &lt;value&gt;2458444610&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     ...   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;ethernet_uid&lt;/name&gt;     &lt;value&gt;3223801380&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;ethernet_uid&lt;/name&gt;     &lt;value&gt;1372827362&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;vm_info&lt;/name&gt;     &lt;value&gt;1294737779&lt;/value&gt;   &lt;/criteria&gt;   ...   &lt;criteria&gt;     &lt;name&gt;sid&lt;/name&gt;     &lt;value&gt;3392305219&lt;/value&gt;   &lt;/criteria&gt; </pre>	<pre> &lt;fingerprint_control_type&gt;ISV Managed &lt;fingerprint_info&gt;   &lt;criteria&gt;     &lt;name&gt;cpu&lt;/name&gt;     &lt;value&gt;3920118414&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;mainboard&lt;/name&gt;     &lt;value&gt;2458444610&lt;/value&gt;   &lt;/criteria&gt;   ...   &lt;criteria&gt;     &lt;name&gt;ethernet_uid&lt;/name&gt;     &lt;value&gt;3223801380&lt;/value&gt;   &lt;/criteria&gt;   &lt;criteria&gt;     &lt;name&gt;ethernet_uid&lt;/name&gt;     &lt;value&gt;1372827362&lt;/value&gt;   &lt;/criteria&gt;   ...   &lt;criteria&gt;     &lt;name&gt;vm_info&lt;/name&gt;     &lt;value&gt;1294737779&lt;/value&gt;   &lt;/criteria&gt;   ...   &lt;criteria&gt;     &lt;name&gt;sid&lt;/name&gt;     &lt;value&gt;3392305218&lt;/value&gt;   &lt;/criteria&gt; </pre>

As seen in the code:

- > The ethernet addresses (ethernet\_uid) in the system and reference fingerprints are identical.
- > The security identifiers (sid) in the system and reference fingerprints are not identical.
- > The value of vm\_info indicates that both fingerprints are from physical machines.

Since the required number of matching criteria is 2, and only one 1 set of criteria match, these fingerprints satisfy the criteria for the custom scheme to indicate that the license is from a cloned machine.



## How to Clear the "Cloned" Status for a Product License

If a Product license is disabled because it has been identified as "cloned", proceed as follows to re-enable the license:

- > For a Product license generated using Sentinel LDK-EMS:
  - a. In Sentinel LDK-EMS, check in the C2V file from the disabled Product license, and click **Clear Clone**. Sentinel LDK-EMS generates a V2C file.
  - b. Return the V2C file to the customer to apply on the machine where the Product license was disabled.
- > For a Product license generated using Sentinel License Generation API: Use the provided function to clear the "cloned" status for the license. For more information, see the [Sentinel License Generation API Reference](#).

**NOTE** To clear clone detection for a Product with a custom clone protection scheme:

- > In cases where the vendor selected two or more clone protection criteria, at least one of the selected criteria must match in the reference and system fingerprints.
- > In cases where the vendor selected only one clone protection criteria, secure storage ID must match in the reference and system fingerprints.

# APPENDIX E: How Sentinel LDK Protects Time-based Licenses With V-Clock

This appendix describes the technology used in Sentinel LDK to prevent a user from extending the duration of a software license that is locked to the V-Clock in a Sentinel protection key by adjusting the computer's system clock.

V-Clock is a virtual clock that is available in Sentinel SL keys and in all types of Sentinel HL (Driverless configuration) keys except for Sentinel HL Basic keys. For Products that are licensed with Sentinel SL keys, V-Clock is always available. For Products that are licensed with Sentinel HL (Driverless configuration) keys, V-Clock must be specifically enabled for each Product.

**NOTE** The use of V-Clock in Sentinel HL Pro keys is only available if your Sentinel LDK Master license contains the V-Clock module.

V-Clock does not provide the same level of control as the real-time clock in Sentinel HL Time keys and Sentinel HL NetTime keys. However, V-Clock prevents the end user from setting the system time back to an earlier date and time, and thus tampering with time-based licenses.

The expiration period or date for a time-based license is initially calculated according to the system clock of the end user's machine.

Sentinel License Manager reads the system time at Sentinel License Manager startup (by default, part of the machine startup). Sentinel License Manager subsequently uses its internal running time to calculate the time. When an application that is protected with V-Clock is executed for the first time, Sentinel License Manager queries its internal clock to determine the start time of the software's license duration.

- > If the license duration is a fixed period (for example, 30 days or 1 year), Sentinel License Manager calculates the actual date on which the license must stop working and the information is stored in the secure storage area of the protection key. The secure storage for a Sentinel SL key is on the hard drive of the end user's computer. The secure storage for a Sentinel HL key is in the HL key.
- > If the license is to expire on a specific date, Sentinel License Manager records that date.

Expiration time is determined using the formula:

`[current Sentinel License Manager time] + number of seconds to expiration`

The information is stored in the secure storage area of the protection key.

**NOTE** The V-Clock time is not automatically updated by the License Manager if you are using only Sentinel Licensing API to protect your application. To update the V-Clock time for a given protection key with Sentinel Licensing API, you must establish a session for the key. The V-Clock time is updated for the Login function, and then for each of the following additional functions: Logout, Read, Write, Encrypt, Decrypt.

## Tampering with the System Clock

If a user resets the system clock of the machine to which the software license is locked:

- > As long as Sentinel License Manager remains active, the changed time does not affect the expiration time of the license, since the calculations are all made within the License Manager, which uses the time of its last startup.
- > If Sentinel License Manager is stopped and restarted (for example: if the machine is rebooted), the License Manager compares its last recorded internal time with the time of the system clock. When Sentinel License Manager detects that the time on the system clock is earlier than that of its internal clock, protected applications with time-based licenses are deactivated. The applications are reactivated automatically when the system clock is equal to or later than the time in the License Manager.

**NOTE** Sentinel License Manager allows the system clock to run up to 24 hours earlier than its internal clock. This accommodates situations where the protected application is used across different time zones.

## Re-enabling a Blocked Protected Application

As indicated above, a blocked protected application is automatically re-enabled when the time on the system clock is no longer earlier than the V-Clock time. The application will be accessible if the license for the application has not yet expired.

Under certain circumstances, you may want to re-enable a blocked application by changing the V-Clock time. This can be accomplished by receiving a C2V file for the protection key from the customer and then returning a V2C file that provides an update to the V-Clock time.

**NOTE** Before applying a V2C file to reset the V-Clock using the system clock, the user should ensure that the system clock is set to the current date and time.

## Setting Fallback to V-Clock If the RTC Battery in a Sentinel HL key is Depleted

If the battery for the real-time clock on a Sentinel HL (Driverless configuration) Time or NetTime key is depleted, the key is no longer accepted for time-based licenses.

You can configure a Sentinel HL Time or NetTime key to switch automatically to the V-Clock if the battery becomes depleted. If the real-time clock on the Sentinel HL key stops operating, protected applications, including those with time-based licenses, will continue to run.

In Sentinel License Generation API, you can implement fallback to V-Clock for a Sentinel HL key by including the tag `<fallback_to_vclock>` in a license definition. In Sentinel LDK-EMS, you can select the global configuration parameter **Fallback to V-Clock** in the Administration Console in order to implement fallback to V-Clock in all generated licenses.

### NOTE

- > Once you have enabled fallback to V-Clock for a Sentinel HL Time or NetTime key, this functionality can be disabled in the key only if the battery is not yet depleted.
- > After the real-time clock stops working, the Sentinel HL key must be disconnected and reconnected in order to switch over to the V-Clock.

# APPENDIX F: How to Optimize Performance for Sentinel LDK Run-time Environment

Certain configuration parameters or activities performed by a protected application can lead to reduced performance in the Sentinel LDK Run-time Environment.

This section describes how you can optimize the environment and protected application to achieve better performance.

## SL UserMode License

---

The presence of an SL UserMode license in a protection key on the end user's computer increases the time required for the first login/get\_info operation performed for a protected application, even if the license is not required for that application. Therefore, do not place an SL UserMode license on a computer unless required.

## Run-time Environment

---

For best performance, ensure that when the Run-time Environment is required, the Run-time Environment on the end user's computer is the most current. In addition, the Run-time Environment provides better performance after it has been active for at least three minutes.

## Testing for Presence of Features

---

To determine whether certain Features exist in a protection key before using them, the protected application can call the **GetInfo** function in the Licensing API. This function can retrieve a list of all the Features that exist in the protection key. This is more efficient than attempting to log in and then log out immediately to individual Features just to determine if they exist. In addition, use of the **GetInfo** function does not consume a license. However, after using the **GetInfo** function, the protected application should call the **Login** function to log in to Features to use.

## APPENDIX G: Upgrading Sentinel HL Keys

The configuration of Sentinel HL keys can be upgraded before or after delivery to customers as follows:

- > Sentinel HL (HASP configuration) keys can be upgraded to Sentinel HL (Driverless configuration) keys.
- > Sentinel HL (Driverless configuration) standalone (non-Net) keys can be converted to Sentinel HL (Driverless configuration) network keys.

Each of these upgrades is described below.

This section also describes ["Differences Between Sentinel HL \(Driverless configuration\) keys and Sentinel HASP keys" on page 354](#).

### Upgrading a Sentinel HL Key to Driverless Configuration

A Sentinel HL (HASP configuration) key that was previously delivered to a customer can be upgraded to a Sentinel HL (Driverless configuration) key in the field. In Driverless configuration, this key will employ HID drivers instead of HASP key drivers. (HID drivers are an integral part of the operating system.) As a result:

- > The key is less subject to issues related to operating system upgrades.
- > The key may no longer require the presence of Sentinel LDK Run-time Environment.

All of the licenses and key memory that existed in the Sentinel HL (HASP configuration) key will continue to exist in the key after the upgrade.

**NOTE** Given the following situation:

- > An application is protected with version 6.3 or 6.4 of Sentinel Licensing API libraries and/or Envelope.
- > The Sentinel HL (HASP configuration) key that licenses the application is upgraded to the Driverless configuration.

The application will work correctly after the upgrade. However, the requirement for the presence of the Run-time Environment does not change.

The tables that follow summarize the requirements for working with HL keys.

## Standalone HL Keys

Version of Licensing API or Envelope used to protect the application (lower of the two)	HASP HL key or Sentinel HL (HASP configuration) key	Sentinel HL (Driverless configuration) key
HASP SRM, Sentinel HASP, or Sentinel LDK v.6.0 or v.6.1	Requires Run-time Environment from same (or later) version that was used to protect the application	Not supported. See the warning below.
Sentinel LDK v.6.3	Requires Run-time Environment from Sentinel LDK v.6.3 or later	
Sentinel LDK v.6.4	Requires Run-time Environment from Sentinel LDK v.6.4 or later	
Sentinel LDK v.7.0 or later	Requires Run-time Environment from Sentinel LDK v.7.0 or later	Under Windows, use of Run-time Environment (from Sentinel LDK v.7.0 or later) is optional.

## Net and NetTime HL Keys

Version of Licensing API or Envelope used to protect the application (lower of the two)	HASP HL key or Sentinel HL (HASP configuration) key	Sentinel HL (Driverless configuration) key
HASP SRM, Sentinel HASP, or Sentinel LDK v.6.0 or v.6.1	On the machine where the HL key is connected: Requires Run-time Environment from same (or later) version that was used to protect the application	Not supported. See the warning below.
Sentinel LDK v.6.3	On the machine where the HL key is connected: Requires Run-time Environment from Sentinel LDK v.6.3 or later	
Sentinel LDK v.6.4	On the machine where the HL key is connected: Requires Run-time Environment from Sentinel LDK v.6.4 or later	
Sentinel LDK v.7.0 or later	On the machine where the HL key is connected: Requires Run-time Environment from Sentinel LDK v.7.0 or later	

The following limitations apply:

- The application must be protected using version 6.4 or later of Sentinel Licensing API libraries and/or Envelope. (For Sentinel HL Net keys and Sentinel HL NetTime keys, use version 7.0 or later.)

- > You must be using version 6.4 or later of Sentinel LDK-EMS or License Generation API to generate the Product that upgrades the HL key. (For Sentinel HL Net keys and Sentinel HL NetTime keys, use version 7.0 or later.)
- > The firmware on the Sentinel HL key will be automatically updated as part of the upgrade process.
- > After upgrade, Sentinel HL (Driverless configuration) keys will not be visible in Admin Control Center if the Run-time Environment is earlier than:
  - version 6.50 (Sentinel LDK v.6.3) — for standalone keys
  - version 6.60 (Sentinel LDK v.7.0) — for Net and NetTime keys

An application that is protected with version 6.3 of Sentinel LDK, Licensing API libraries and/or Envelope will work correctly after the Sentinel HL (HASP configuration) key that licenses the application is upgraded to the Driverless configuration. However, the requirement for the presence of the Run-time Environment does not change.

**\*\*WARNING\*\***

**An application that is protected with version 6.1 or earlier of Sentinel LDK libraries, Licensing API libraries and/or Envelope will stop working if the Sentinel HL (HASP configuration) key that licenses the application is upgraded to the Driverless configuration.**

**The upgrade process for the Sentinel HL key is not reversible.**

## Upgrade Requirements

The machine that is used to upgrade a Sentinel HL (HASP configuration) key to a Sentinel HL (Driverless configuration) key must contain a Sentinel LDK Run-time Environment that satisfies the following requirements:

Sentinel HL (HASP configuration) key to upgrade	Required Run-time Environment
Standalone key that contains license information (Features and Products)	Version 6.56 or later
Net or NetTime key that contains license information (Features and Products)	Version 6.60 or later
Any HL key that contains no license information (Features and Products) AND the license update used to upgrade the key contains no license information (Features and Products). Both the key and the license update can contain memory data.	No special version requirements



# Upgrade Process

## To upgrade a Sentinel HL (HASP configuration) key to Sentinel HL (Driverless configuration) key:

- > Create a Base Product or Modification Product that contains the **Upgrade to Driverless** attribute. The Product can be created exclusively to upgrade the Sentinel key, or the **Upgrade to Driverless** attribute can be included in a Product that licenses or modifies the license for a protected application. Apply the Product to the Sentinel HL (HASP configuration) key to be upgraded.

The **Upgrade to Driverless** attribute is ignored if it applied to Sentinel HASP keys or to Sentinel HL (Driverless configuration) keys. Similarly, the attribute is ignored if is applied to an SL AdminMode key, SL UserMode key, or SL Legacy key. No error message is generated.

The Product that contains the **Upgrade to Driverless** attribute can be created using Sentinel LDK-EMS, Sentinel LDK-EMS Web Services, or Sentinel License Generation API.

## To upgrade a Sentinel HL Basic key from HASP configuration to Driverless configuration:

- > On the machine where the Sentinel HL Basic key is connected, use RUS to collect information regarding the key. Use the resulting C2V file with Sentinel License Generation API to generate a V2C file that uses the **Upgrade to Driverless** attribute to upgrade the key.

Apply the V2C file to the Sentinel HL Basic key to be upgraded.

# Converting a Sentinel HL Standalone Key to a Network Key

This topic does not apply to Sentinel HL Basic keys.

The table that follows describes the terminology used in this section.

Term	Description
Sentinel HL standalone key	Any Sentinel HL (Driverless configuration) key other than Net or NetTime keys.
Sentinel HL concurrency-enabled key	A Sentinel HL standalone key that has been updated to support concurrency licenses.
Sentinel HL network key	Any Sentinel HL key that supports network seat licenses. This can be a Net or NetTime key, or a Sentinel HL concurrency-enabled key.

Sentinel HL standalone keys can be updated, before or after delivery to end users, to Sentinel HL concurrency-enabled keys, and thus provide practically the same network functionality as Sentinel HL Net or NetTime keys.

The only difference between a Sentinel HL concurrency-enabled key and a Sentinel HL Net or NetTime key is the manner in which you are charged for network seat licenses. Each Net or NetTime key is provided with a number of network seat licenses, based on the type of key. For HL concurrency-enabled keys, network seat licenses that you provide to your customers are deducted from the **HL Pool of Seats** in your Sentinel LDK Master license. This is similar to the way network seats are charged for Sentinel SL keys.

You update a Sentinel HL standalone key to a Sentinel HL concurrency-enabled key simply by assigning concurrency to a Feature on the key. When this occurs, the License Manager checks the firmware version of the key and may upgrade the firmware. (For more information on firmware upgrades, see the description of firmware in the [Sentinel LDK Release Notes](#).)

The conversion can only occur if License Manager v.7.3 or later is present on the machine where the Sentinel HL key is connected.

**NOTE** When you update a Sentinel HL standalone key to a Sentinel HL concurrency-enabled key, you must also ensure that the Sentinel LDK Run-time Environment is installed on the machine where the key is connected. For more information, see ["Situations That Require Sentinel LDK Run-time Environment" on page 194](#).

Note the following:

- > Feature ID 0 for a Sentinel HL concurrency-enabled key shows the key as a NET key with unlimited concurrency as long as any other Feature on the key requires concurrency. If the requirement for concurrency are removed from the key, Feature ID 0 will show the key as a standalone key.
- > In Sentinel License Generation API, you can prevent the upgrade of the Firmware for a Sentinel key when you update a license. However, if the existing Firmware on the key does not support the functionality in the update you are attempting to perform, the update will fail as a result. For more information, see the [Sentinel License Generation API Reference](#).
- > All Sentinel HL (Driverless configuration) key (except for HL Basic keys) will be shown as capable devices for licenses that require concurrency.

Sentinel HL (HASP configuration) standalone keys can be upgraded and updated to Sentinel HL (Driverless configuration) concurrency-enabled keys in a single update operation. Upgrade the key to the Driverless configuration as described earlier in this appendix, and at the same time assign concurrency to a Feature on the key.

## Differences Between Sentinel HL (Driverless configuration) keys and Sentinel HASP keys

There is differences between Driverless keys and HASP keys when applying license update files (V2C or V2CP files, all referred to below as *V2C updates*).

- > For HASP keys:

Sentinel LDK does not support applying a given V2C update multiple times. If the process of applying a V2C update to a HASP key is interrupted before completion, an additional attempt to apply the same V2C update will fail.

> For Driverless keys:

A given V2C update can be applied multiple times. After the first successful application of the V2C update, any additional attempts to apply the V2C update are simply ignored. If the process of applying a V2C update to a Driverless key is interrupted before completion, you can attempt to apply the same V2C update additional times until it succeeds.

However: If the customer uses the same C2V file to generate two different V2C updates, then after applying the first V2C update, the second V2C update might be applied partially to the Driverless key. Thales recommends that customers make sure that a given C2V file is only submitted to the vendor once to avoid this issue.

## APPENDIX H: Protecting Applications in Docker Containers

Sentinel LDK supports protected applications that execute in Docker containers, within the limitations described in this appendix.

The Product license for a protected application that runs in a Docker container can be deployed using either HL keys or using SL keys, as described below.

### NOTE

- > This appendix is applicable only for Docker containers under Linux unless except where stated otherwise.
- > This appendix is applicable for applications that are protected and licensed using Sentinel LDK v.7.10 and later.

*In this appendix:*

- > ["Using SL Keys" on the next page](#)
- > ["Using HL Keys" on page 358](#)
- > ["Additional Considerations" on page 359](#)
- > ["Side-By-Side Comparison" on page 360](#)

## Using SL Keys

Sentinel LDK supports the use of SL keys for protected applications that execute in a Docker container. The Run-time Environment can be installed on the host machine or within the Docker container.

The Run-time Environment and the SL key for a protected application that runs in a Docker container can be configured using one of the options described below.

### > Option 1 - Outside of Container

Type of Key	Location of Run-time Environment	Location of SL Key
SL AdminMode key (Includes cloud licensing)	Host machine or remote machine	Host machine or remote machine

The RTE and SL key are installed outside of the Docker container. This option does not have any limitations. The RTE works as usual. The protected application running in the Docker container accesses the license via network communication.

Using this option, a Linux container can be on either a Linux or Windows host.

This option relies on network connectivity, removing dependency on the host operating system. This option is different from local installation of an SL key, which requires Sentinel LDK to manage a local secure storage. As long as the application is running under an operating system that Sentinel LDK supports for consuming network licenses, the host environment (in the case of virtual machine or virtualized container) is of no importance.

If the host machine is a physical machine, you can prevent installation of SL AdminMode keys in the container by disabling support for virtual machines when you create the keys.

Thales recommends the use of Option 1 in the following scenarios:

- One or more hosts in a local network (an SL key with concurrency)
- Single host in the cloud (CL key) with the license configured with "Count Each Station" to define what is to be counted as a concurrent instance.

### > Option 2 - Within Container

Type of Key	Location of Run-time Environment	Location of SL Key
SL AdminMode key SL UserMode Key	Within the container (RTE version must be 7.100 or later.)	Within the container

You do not need to install anything on the host. However, you can only use this option with perpetual licenses that do not have a concurrency limit. Any other type of software licenses will be regarded as a clone the next time the container is restarted.

With this option, Thales recommends that you install the license at every container startup. This can be accomplished by simply placing the V2C file in the appropriate directory.

You should not save the container image after secure storage has been created. If the container image is saved, the secure storage would be regarded as "restored manually" at container startup and would be completely recreated automatically. This slows down the startup process.

### > Option 3 - Mixed Solution

Type of Key	Location of Run-time Environment	Location of SL Key
SL AdminMode key (Includes cloud licensing)	Within the container (RTE version must be 7.100 or later.)	Host machine or remote machine

To use the mixed solution, perform the procedures that follow.

#### Preparation:

- a. Install the RTE inside the Docker container OR on the host machine.
- b. Use the License Manager in the RTE to install and activate the SL key on the host machine.
- c. If you installed the RTE on the host machine:
  - i. Uninstall the RTE from the host machine.
  - ii. Install the RTE inside the Docker container.

#### Consuming licenses:

An application should consume a license from the SL key using the License Manager from the RTE installed inside the Docker container.

#### Additional considerations for the mixed solution:

You install the RTE inside the container, but configure Docker to keep the license storage directories on the host to be able to install any kind of license.

**NOTE** You cannot have a License Manager service running both inside the container and on the host. When using this option, ensure that the License Manager service executes only inside the container.

The host machine or remote machine (service) can supply a mounted persistence volume as SL storage. In a cloud environment, persistence volume is a resource that is backed by a persistent disk or volume service.

You can configure Docker to keep the license storage directories on the host using the Docker **-v** option. For example: The following command starts the "ubuntu" container, keeping the `/var/hasplm` and `/etc/hasplm` directories on the host.

```
$ sudo docker run -it -v /var/hasplm:/var/hasplm -v /etc/hasplm:/etc/hasplm -p 1947:1947 ubuntu
```

You can then install the RTE inside the container. (If the license storage directories already exist and contain licenses, you will be able to access these licenses from inside the container.)

## Using HL Keys

Sentinel LDK supports the use of HL keys for protected applications that execute in a Docker container.

When installing Sentinel LDK Run-time Environment (*RTE*) for use with HL keys, the RTE can be installed either on the host machine or within the Docker container.

### > Option 1 - Outside of Container

Location of Run-time Environment	HL Key Access
Host machine	HL key accessed from the host machine

The protected application running in the Docker container accesses the license on the HL key via network communication. Only network licenses are supported.

Thales recommends that you use this option if the license supports remote access. Access the HL key via the RTE and not directly from the Docker container.

### > Option 2 - Within Container

Location of Run-time Environment	HL Key Access
Within the container (RTE version must be 7.100 or later.)	HL key accessed from inside the container

This includes a scenario in which the Licensing API accesses the HL keys directly, without the need for the RTE.

When the RTE is installed within the Docker container, the host must be configured to share all USB devices with the container. You can accomplish this by issuing the following command on the host machine:

```
$ sudo docker run -it --device /dev/bus/usb:/dev/bus/usb ubuntu
```

It is also possible to share only the specific HL key by specifying the key's path, but you must implement some logic to identify this path. For example:

```
$ sudo docker run -it --device /dev/bus/usb/003/008:/dev/bus/usb/003/008 ubuntu
```

## Additional Considerations

---

- > Distribution of Docker images should be done before any license is installed. License activation should then be done after the user chooses the host they want to use. Distributing this container to other hosts will render this license unusable (regarded as cloned).
- > An SL license of any type other than **Perpetual** that is installed in the container become a cloned license the next time the container restarts. To prevent this, do one of the following:
  - Use one of the other installation options. The best option is to configure the secure storage on the host before installing the license.
  - Request the user to provide a C2V from the host and install a license with concurrency of 1.
- > When the License Manager operates in a Docker container: To configure the License Manager and establish communication between the License Manager and the protected application, you may need to configure the License Manager INI file. For more information, see ["Working Directly With License Manager Configuration Files" on page 221](#).

## Side-By-Side Comparison

The tables that follow provides a side-by-side comparison of the different licensing options described in this appendix.

### SL Key Options

	Option 1: Outside of Container	Option 2: Within Container	Option 3: Mixed Solution
<b>Description</b>	CL key with concurrency on a remote machine or Docker host; supports cloud licensing if the host/container is in the cloud. SL keys (not cloud-enabled) are suitable if the host/container is in a local network.	SL key and RTE inside the container	SL key on the host and shared with container; RTE inside the container
<b>Type of Key</b>	SL AdminMode key	SL AdminMode key SL UserMode key	SL AdminMode key and SL UserMode key
<b>Docker Host Requires Configuration?</b>	No	No	Yes
<b>Move the Container to Another Host Without Reactivation?</b>	Yes	No	No

### Supported License Types and License Actions



	Option 1: Outside of Container	Option 2: Within Container	Option 3: Mixed Solution
<b>Perpetual</b>	Yes	Yes	If the license is activated from a License Manager inside the Docker container, this option is identical to option 2.
<b>Expiration Date</b>	Yes	No <sup>1</sup>	
<b>Execution Count</b>	Yes	No <sup>1</sup>	
<b>Time Period</b>	Yes	No <sup>1</sup>	If the license is activated from a License Manager on the host, this option is identical to option 1.
<b>Concurrency</b>	Yes	No <sup>1</sup>	
<b>Detach</b>	Yes	Not applicable	
<b>Rehost</b>	Yes (to another remote machine)	No	

<sup>1</sup>This configuration will be blocked during license generation. Use option 1 with a single network seat or use option 3 with the license activated from the host.

## HL Key Options

	Option 1: Outside of Container	Option 2: Within Container
<b>Description</b>	HL key with concurrency on a remote machine or Docker host	HL key (no concurrency) connected to the host and shared with the container
<b>Type of Key</b>	HL key	HL key
<b>Anything Located Outside of the Container?</b>	Yes (RTE on remote machine)	Only Physical key
<b>Docker Host Requires Configuration?</b>	No	Yes
<b>Move the Container to Another Host Without Reactivation?</b>	Yes	Yes

## Supported License Types and License Actions

	Option 1: Outside of Container	Option 2: Within Container
Perpetual	Yes	Yes
Expiration Date	Yes	Yes
ExecutionCount	Yes	Yes
Time Period	Yes	Yes
Time from License Generation	Yes	Yes
Concurrency	Yes	Yes
Detach	No	Not applicable
Rehost	Move the key	Move the key

## APPENDIX I: Protecting Applications in Linux LXC Containers

Sentinel LDK supports protected applications that execute in LXC containers under Linux, within the limitations described in this appendix.

The Product license for a protected application that runs in an LXC container can be deployed using either HL keys or using SL keys, as described below.

**NOTE** This appendix is applicable for applications that are protected and licensed using Sentinel LDK v.8.2 and later.

*In this appendix:*

- > ["Using SL Keys" on the next page](#)
- > ["Using HL Keys" on page 364](#)

## Using SL Keys

Sentinel LDK supports the use of SL keys for protected applications that execute in a LXC container. The Run-time Environment can be installed on the host machine or within the LXC container.

The Run-time Environment and the SL key for a protected application that runs in a LXC container can be configured using one of the options described below.

### > Option 1 - Outside of Container

Type of Key	Location of Run-time Environment	Location of SL Key
SL AdminMode key (Includes cloud licensing)	Host machine or remote machine	Host machine or remote machine

The RTE and SL key are installed outside of the LXC container. This option does not have any limitations. The RTE works as usual. The protected application running in the LXC container accesses the license via network communication.

If the host machine is a physical machine, you can prevent installation of SL AdminMode keys in the container by disabling support for virtual machines when you create the keys.

### > Option 2 - Within Container

Type of Key	Location of Run-time Environment	Location of SL Key
SL AdminMode key SL UserMode Key	Within the container (RTE version must be 8.21 or later.)	Within the container

You do not need to install anything on the host.

### > Option 3 - Mixed Solution

Type of Key	Location of Run-time Environment	Location of SL Key
SL AdminMode key (Includes cloud licensing)	Within the container (RTE version must be 8.21 or later.)	Host machine or remote machine

You install the RTE inside the container, but configure LXC to keep the license storage directories on the host to be able to install any kind of license.

**NOTE** You cannot install the RTE both inside the container and on the host. When using this option, ensure that the RTE executes only inside the container.

The host machine or remote machine (service) can supply a mounted persistence volume as SL storage. In a cloud environment, persistence volume is a resource that is backed by a persistent disk or volume service.

You can configure LXC to keep the license storage directories on the host using the LXC config file from `/var/lib/lxc/<containername>/`. Add the following commands to the **config** file, keeping the `/var/hasplm` and `/etc/hasplm` directories on the host.

```
lxc.mount.entry=/var/hasplm var/hasplm none bind,optional,0
lxc.mount.entry=/etc/hasplm etc/hasplm none bind,optional,0
```

# Using HL Keys

Sentinel LDK supports the use of HL keys for protected applications that execute in a LXC container.

When installing Sentinel LDK Run-time Environment (RTE) for use with HL keys, the RTE can be installed either on the host machine or within the LXC container.

## > Option 1 - Outside of Container

Location of Run-time Environment	HL Key Access
Host machine	HL key accessed from the host machine

The protected application running in the LXC container accesses the license on the HL key via network communication. Only network licenses are supported.

Thales recommends that you use this option if the license supports remote access. Access the HL key via the RTE and not directly from the LXC container.

## > Option 2 - Within Container

Location of Run-time Environment	HL Key Access
Within the container (RTE version must be 8.21 or later.)	HL key accessed from inside the container

This includes a scenario in which the Licensing API accesses the HL keys directly, without the need for the RTE.

When the RTE is installed within the LXC container, the host must be configured to share the specific HL key. For example:

**a.** List USB key

```
$ lsusb
Bus 002 Device 006: ID 0529:0003 Aladdin Knowledge Systems
```

**b.** The device node is visible as:

```
$ ls -l /dev/bus/usb/002/006
crw-rw-rw- 1 root plugdev 189, 133 Apr  1 08:06 /dev/bus/usb/002/006
```

**c.** Add cgroup permissions to LXC config file (*/var/lib/lxc/<containerName>/config*):

```
lxc.cgroup.devices.allow = c 189:* rwm
lxc.mount.entry=/dev/bus/usb/002 dev/bus/usb/002 none bind,optional,create=dir
```

# APPENDIX J: Troubleshooting

The first part of this appendix provides a checklist to help you solve some of the most common problems that your customers might encounter when using the Sentinel HL keys. The second part lists specific problems you or your customers may experience, together with the solutions.

Sentinel HL keys conform to the highest standards of quality assurance. However, like any other PC peripheral device, a Sentinel HL key might not operate on certain PC configurations because of faulty equipment or improper installation. This appendix can help you in such a situation.

In addition to the information in this appendix, you can access the Sentinel Knowledge Base at:

[https://supportportal.thalesgroup.com/csm/?id=kb\\_home\\_page](https://supportportal.thalesgroup.com/csm/?id=kb_home_page)

The Knowledge Base contains a comprehensive listing of solutions to general and specific problems.

To avoid potential difficulties, ensure you are using current Sentinel LDK software versions. Contact your local Thales representative for the latest updates, or visit the Thales downloads page at:

<https://cpl.thalesgroup.com/software-monetization/sentinel-drivers>

## Checklist

If a customer reports a problem, check the following:

- > What the returned error code or message says. For additional information, see the status codes in the [Sentinel License Generation API Reference](#).
- > Whether a Sentinel HL key is connected correctly to the USB port.
- > Whether your customer’s hardware or the operating system indicates technical malfunction, such as device manager collisions, system events, bootlog failures, or other issues.
- > Whether Sentinel Admin Control Center can access the Sentinel HL key.
- > Whether the problem occurs when the protected application runs on another PC of the same model.

## Problems and Solutions

Problem	Sentinel HL key drivers do not install.
---------	---

<b>Solution</b>	Are older Sentinel HL key drivers installed on the machine? Uninstall the older driver using the installer corresponding to the older driver version. For additional information, see the Sentinel HL key driver documentation. After the older drivers are removed, install the Sentinel HL drivers. For additional information, see the <a href="#">Sentinel LDK Installation Guide</a> .
<b>Problem</b>	The protected application cannot find a Sentinel HL key.
<b>Solution</b>	<p>Does the Sentinel HL key LED light up? If not, this could be for one of the following reasons:</p> <ul style="list-style-type: none"> <li>&gt; The key is not connected properly to the USB port. Disconnect, then reconnect after a few seconds. If the LED lights, the application should be able to access the key.</li> <li>&gt; The required Sentinel HL key drivers are not installed. If you are running Sentinel LDK on a Windows platform, check for an entry for Sentinel LDK in the Device Manager utility. If there is no entry, you must install the drivers using one of the methods in the <i>Sentinel LDK Installation Guide</i>.</li> <li>&gt; Check if the USB port is functioning correctly. Disconnect all other USB devices from their respective ports. Connect the Sentinel HL key to a different USB port. Try using a different USB device in the port from which the Sentinel HL key was not accessible.</li> <li>&gt; Open the Windows Services window and check that Sentinel License Manager is running.</li> <li>&gt; Check that the Batch Code on the Sentinel HL key matches the Batch Code of the protected application.</li> </ul>
<b>Problem</b>	Web pages for Admin Control Center do not display in your Web browser on a Windows machine.
<b>Solution</b>	<p>Check the following:</p> <ul style="list-style-type: none"> <li>&gt; Confirm that the Sentinel LDK License Manager service is active.</li> <li>&gt; Some other program that you installed may have incorrectly installed special TCP/IP drivers. As a result, WinSock configuration may be damaged. To resolve this problem, run the command <code>netsh winsock reset</code> from an Administrator shell, and then restart the machine.</li> </ul>
<b>Problem</b>	The application takes a long time to find the Sentinel protection key on a large network.
<b>Solution</b>	It is recommended that you customize the search mechanism. Use Admin Control Center configuration to specify a search criteria, and to define the server addresses to be searched. By doing so, the Admin Control Center searches for the Sentinel protection key at a specific address, which is much faster.

<b>Problem</b>	You receive an error message indicating that Sentinel License Manager was not found.
<b>Solution</b>	<p>The error message might be for one of the following reasons:</p> <ul style="list-style-type: none"> <li>&gt; Sentinel License Manager was not loaded. Try restarting Sentinel License Manager in the Windows Services window.</li> <li>&gt; There is a communication error with the machine on which the Sentinel protection key is located. If you repeatedly receive the error message, try using a different search mechanism.</li> </ul>
<b>Problem</b>	You cannot add files when using the Sentinel LDK Data Protection utility.
<b>Solution</b>	<p>The problem may occur for one of the following reasons:</p> <ul style="list-style-type: none"> <li>&gt; You are attempting to add a list that includes problematic files. Remove all problematic files marked in red in the File list.</li> <li>&gt; You are attempting to add a file that is outside the scope of the filters defined in Sentinel Envelope. You must protect your software again using the new file filter settings.</li> <li>&gt; For additional information, see <a href="#">"Protecting Data Files" on page 77</a>.</li> </ul>
<b>Problem</b>	When using Sentinel LDK Data Protection utility, you receive a message that no data filters were defined for a program in a Sentinel Envelope project.
<b>Solution</b>	The problem cannot be solved using the Data Protection utility. You need to use Sentinel LDK Envelope to protect your software again, and to specify file filter settings.
<b>Problem</b>	When running a protected Java application, error code 22 is thrown.
<b>Solution</b>	When a java application is protected using both Sentinel LDK Envelope and Sentinel Licensing API, the JAR files used by the protected application must have the same version number. Ensure that you are using both Sentinel LDK Envelope and Sentinel Licensing API from the same version of Sentinel LDK.
<b>Problem</b>	When you are logged in to both the Vendor Portal and the Customer Portal simultaneously in Sentinel LDK-EMS in different tabs of a single Web browser instance, a message for an internal error is generated.



<b>Solution</b>	To work simultaneously with the Sentinel LDK-EMS Vendor Portal and Customer Portal on a single machine, use two separate Web browser instances.
<b>Problem</b>	When you download a C2V file (for example, abc.C2V) from Admin Control Center using the Safari Web browser, Safari appends ".xml" to the file name (for example, abc.c2v.xml). As a result, Sentinel LDK-EMS blocks the file.
<b>Solution</b>	Before you attempt to use the C2V file in Sentinel LDK-EMS, remove the ".xml" extension from the file name.
<b>Problem</b>	When an application calls a Sentinel LDK-EMS Web Service, error 403 is returned.
<b>Solution</b>	<p>HTTP verb tampering is an attack that exploits HTTP verb-based authentication and access control mechanisms. The vulnerability lies in the configuration of the security constraints that specify HTTP verbs that often allow more access than intended.</p> <p>An application's authentication and authorization mechanisms can be bypassed with HTTP verb tampering when the security control fails to block verbs that are not listed.</p> <p>Effective with version 7.9, Sentinel LDK-EMS Web Services have been modified to prevent HTTP verb tampering. The methods that are used to call Web Services have been whitelisted: GET, POST, DELETE, PUT. When calling these methods in your application, you must ensure that they are written using only upper-case letters as shown here. Variations that used lower-case or mixed-case letters have been blacklisted.</p> <p>This is a standard practice as suggested by the Internet community and is mentioned in RFC standard protocols (RFC 7230 and 7231).</p> <p>The use of method calls that do not comply with these practices will cause your application to fail with error code 403.</p>
<b>Problem</b>	<p>Using Sentinel Licensing API, you call the GetInfo function in the to fetch the fingerprint for an SL UserMode key. The fingerprint that is retrieved resembles the following:</p> <pre>&lt;?xml version="1.0" encoding="UTF-8" ?&gt; &lt;hasp_info&gt;   &lt;host_fingerprint type="SL-UserMode" vendorid="37515"     crc="3820837874"&gt;NHhJSQ==&lt;/host_fingerprint&gt; &lt;/hasp_info&gt;</pre> <p>This indicates that the API function retrieved an empty fingerprint. This occurs when the SL UserMode license cannot be accessed or is corrupted.</p>

**Solution**

The most common reasons for this issue is that access to the directory that contains the SL UserMode licenses is denied. When you resolve the access permission issue, the API function will be able to fetch the fingerprint as required.

If the problem is not caused by access permissions, the most likely cause is that the SL UserMode licenses have become corrupted.

To resolve this issue, you must clear the entire SL UserMode license directory. As a result, all locked licenses will be invalidated due to clone protection. It will be necessary to reissue these licenses.

Unlocked licenses can be reinstalled in the directory.

For information on the location of SL UserMode licenses on the user's machine, see ["Viewing License Updates" on page 134](#).

## APPENDIX K: Choosing Between Sentinel LDK-EMS and Sentinel EMS

Sentinel LDK is available with your choice of two different entitlement management systems. An *entitlement management system* is a back-office server application that enables you to define, produce and update Sentinel protection keys and to define and process entitlements (customer orders).

- > **Sentinel LDK-EMS** is a dedicated legacy entitlement management system for Sentinel LDK. This system is available either for installation on the vendor's server (on-premises) or as a service (hosted on Thales servers). Sentinel LDK-EMS provides a user interface and REST API to manage resources.
- > **Sentinel EMS** is an improved, enterprise-level system that integrates with Sentinel LDK. Sentinel EMS supports multiple enforcement types, including custom and third-party enforcements. Sentinel EMS is available as a service that is hosted on Thales servers, where it is enhanced continuously with improvements and updates. Sentinel EMS provides an advanced user interface and REST API to manage resources.

For a detailed comparison of the functionality and terminology used in these systems, see [Comparison Between Sentinel LDK-EMS and Sentinel EMS](#).

Need more information? To determine which entitlement management system is right for you, contact your Thales representative.

# Glossary

<b>Activation</b>	The process in which an SL key is locked to a specific computer or a license is burned to an HL key. Following activation, the protected software can be used on the end user's computer according to the activated license.
<b>Activation counter</b>	Licensing element that indicates the number of times that a Feature, which is licensed using Sentinel LDK, can be run.
<b>Admin License Manager</b>	Standalone License Manager that is implemented as a service in Windows or as a daemon in Mac and Linux Intel/ARM. The Admin License Manager handles communication between the protected application and the protection keys. The Admin License Manager is installed as part of Sentinel Run-time Environment, together with Sentinel Admin Control Center. See also: <a href="#">"License Manager" on page 377</a>
<b>AES</b>	Advanced Encryption Standard (AES) algorithm that is the basis for the Sentinel LDK encryption and decryption.
<b>Anti-debugging</b>	Measures that are applied by the Sentinel LDK system to block potential attacks intended to undermine the protection scheme.
<b>API samples</b>	Sample applications that utilize the Sentinel Licensing API. A learning tool used for implementing the Sentinel Licensing API.
<b>AppOnChip</b>	A protection functionality in Sentinel LDK Envelope that moves the execution of selected code fragments from the protected application to the Sentinel HL (Driverless configuration) key. This enhances the security of the protected application.
<b>Automatic Detach</b>	When you start to work with a protected application, a license can be automatically detached from the SL key and attached to your machine. The license is valid for the number of hours specified in Admin Control Center. This enables you to continue working even if your machine loses its connection to the SL key. See also: <a href="#">"Detach" on page 375</a>
<b>Background checks</b>	Random checks executed by protected applications for a required Sentinel protection key.

<b>Backward compatibility</b>	Ability to share data or commands with applications protected with earlier versions. Sentinel LDK backward compatibility includes the ability to read and write data, set real-time clocks, and process other 'legacy' commands.
<b>Base Product</b>	An original Product that has been created from scratch from which other Products may be created. All Modification Products, Unlocked Products and Cancellation Products are created from Base Products.
<b>Batch Code</b>	Unique character string that represents a Vendor Code. The Batch Code is used in defining Features, Products and orders. It is also used for ordering Sentinel protection keys. With Sentinel HL keys, the code is printed on the Sentinel HL key label.
<b>Bundle (of Unlocked Products)</b>	A program that you create that installs a V2C file containing one or more Unlocked Product licenses, your Vendor libraries and a customized Run-time Environment installer. When this package is installed together with your protected application or applications, the applications are ready for immediate execution; no additional processing is required to activate the licenses. Bundles are useful for installing trialware or for installing software that should be protected but does not have licensing requirements.
<b>C2V file</b>	Customer-to-Vendor file. This is a file that is sent by the customer to the vendor, containing data about deployed Sentinel protection keys or data about the customer's computer. C2V files can be safely sent using regular email. See also: <a href="#">"V2C file" on page 382</a>
<b>Cancellation Product</b>	A Product that cancels the licensing details of another Product. A Cancellation Product can be used to revoke a deployed license, or to remove a license from a specified computer so that it can be transferred to another computer.
<b>Channel Partner</b>	A company that partners with you to market and sell your products. Sentinel LDK-EMS enables you to allow your channel partners to access Sentinel LDK-EMS functionality to assist them in servicing their customers.
<b>Channel Partner user</b>	A Sentinel LDK-EMS user who is associated with a specific channel partner. This type of user can perform a limited range of activities for associated end-user customers.
<b>CL key</b>	A cloud-enabled SL AdminMode key (that is, when the SL key was generated, the vendor's Sentinel LDK Master license contained an active Cloud Licensing module). To use a CL key for cloud licensing, licenses must have concurrency enabled.

<b>Cloud licensing</b>	A licensing scheme that enables end users to access local software with a license hosted in the cloud. Cloud licensing uses identity-based access to give the vendor or the customer granular control over who can access a network seat from a license.
<b>Concurrency</b>	<p>A licensing attribute that allows a single protection key to be used by one or more instances of a protected application running on different computers in a network. In a Product, concurrency is defined for each Feature license type.</p> <p>Each instance of the protected application that can be used simultaneously is referred to as a <i>network seat</i> (or a <i>floating license</i>).</p> <p>Management of the license in the network is controlled using the Sentinel License Manager.</p> <p>For more information about concurrency, see <a href="#">"Specifying the License Terms for Features in a Product" on page 110</a>.</p>
<b>Customer Portal</b>	A Web portal in Sentinel LDK-EMS that can be accessed by customers. This portal is used to activate licenses or to automatically apply updates to the protection keys on the customer's machine.
<b>Cross-locking</b>	Indicates that protection can be applied to both Sentinel HL and Sentinel SL keys.
<b>Data Protection utility</b>	Utility for encrypting and (optionally) licensing data files that are accessed by programs protected by Sentinel LDK Envelope. (Formerly DataHASP)
<b>Decryption</b>	The process of decrypting data that has been encrypted.
<b>Default Feature</b>	<p>A Feature (with Feature ID "0") that is always available in a Sentinel protection key and can be used to provide copy protection without the need to fulfill a Sentinel LDK license. This feature is always perpetual and cannot be modified to use other licensing terms.</p> <p>When you protect an application with Envelope, Feature ID 0 is applied by default if you do not choose any other Feature ID for licensing the application. To license a specific Feature, always apply the relevant Feature ID.</p> <p>If your application will be distributed with Sentinel HL Basic keys, you can use only the default Feature (Feature ID 0) to protect your program.</p>
<b>Demo Vendor Code</b>	See <a href="#">"DEMOMA" below</a> .
<b>DEMOMA</b>	Batch Code used for evaluation purposes with any Sentinel LDK application. Its corresponding Vendor Code is available in the VendorCodes folder of your Sentinel LDK installation.

<b>Detach</b>	<p>Temporarily remove a license from a network pool on a host machine for attachment to a remote recipient machine.</p> <p>See also: <a href="#">"On-demand Detach" on page 378</a>, <a href="#">"Automatic Detach" on page 372</a></p>
<b>Developer key</b>	<p>A special-purpose Sentinel HL key that contains the confidential codes assigned to you by Thales and, optionally, certain Sentinel LDK Master license modules for advanced Sentinel LDK features. The key is used by the software engineers when protecting applications or data files using Sentinel LDK.</p>
<b>Encryption</b>	<p>Translation of data into a confidential code. To read an encrypted file, you must have the correct encryption engine for decrypting the file.</p>
<b>Encryption engine</b>	<p>Encryption engine in a Sentinel protection key—based on the AES algorithm.</p>
<b>Encryption key</b>	<p>Key used with Sentinel Envelope to encrypt a data file.</p>
<b>Encryption level</b>	<p>Number of iterations that the Sentinel Envelope executes with the Sentinel protection key for each interaction.</p>
<b>Entitlement</b>	<p>A request for items to be shipped to a customer. The entitlement specifies one of the following:</p> <ul style="list-style-type: none"> <li>&gt; One or more Product licenses to be applied to Sentinel protection keys.</li> <li>&gt; An update to a protection key, specifying changes to be made to the license terms or data stored in one or more deployed Sentinel protection keys.</li> </ul>
<b>Envelope</b>	<p>See <a href="#">"Sentinel LDK Envelope" on page 380</a>.</p>
<b>Expiration date</b>	<p>Date after which a protected application or Feature stops running.</p>
<b>External License Manager</b>	<p>License Manager that can be provided for each protected application (to replace the <a href="#">"Integrated License Manager" on the next page</a>). Handles communication between the application and the protection key. This License Manager can be upgraded by simply replacing a standalone file.</p> <p>See also: <a href="#">"License Manager" on page 377</a></p>
<b>Feature</b>	<p><b>For software applications:</b> An identifiable functionality that can be independently controlled by a license. In Sentinel LDK, a Feature may be an entire application, a module or a specific functionality such as Print, Save or Draw.</p> <p><b>For data files:</b> A specific Feature can be assigned to an individual data file or to a collection of data files. This enables the vendor to easily manage the licensing of data files.</p>

<b>Feature ID</b>	Unique identifier for a Sentinel LDK-protected <a href="#">Feature</a> . See also <a href="#">Default Feature</a> .
<b>Feature ID 0</b>	See <a href="#">"Default Feature" on page 374</a> .
<b>File filter</b>	File mask that is defined in Sentinel LDK Envelope for a protected application. The file filter is used by the protected application to determine which data files should be handled as encrypted files.
<b>Grace period</b>	An initial period of time (typically 30 to 90 days) or number of executions (typically 30) during which a Product can be used without a Sentinel protection key. See also: <a href="#">"Unlocked Trialware Product" on page 382</a>
<b>H2H file</b>	Host-to-Host file. This file is used to rehost (transfer) a protection key from one end user's machine to another end user's machine.
<b>H2R file</b>	Host-to-Recipient file. This file contains one or more detached Product licenses for temporary attachment to a recipient machine.
<b>Handle</b>	Unique identifier for accessing the context of a Sentinel LDK login session.
<b>HASP</b>	A legacy term that is used to refer to Sentinel protection keys in the HASP and LDK family of products. This term is used in the following contexts: <ul style="list-style-type: none"> <li>&gt; HASP HL keys. Legacy hardware protection keys, now replaced by Sentinel HL keys.</li> <li>&gt; HASP SL keys. Previous name for the software-based Sentinel SL Legacy keys.</li> <li>&gt; HASP_ prefix / namespace. Used in the Sentinel Licensing API.</li> <li>&gt; HASPUserSetup.exe. GUI-based Run-time installer that supports multiple key types (Sentinel HL, HASP HL, HASP4, and Hardlock).</li> <li>&gt; haspdinst.exe. Command-line based Run-time installer similar to HASPUserSetup.exe.</li> </ul>
<b>HASP ID</b>	See <a href="#">"Key ID" on the next page</a> .
<b>HL key</b>	See the various entries for Sentinel HL key
<b>Integrated License Manager</b>	License Manager that is integrated into each protected application. Handles communication between the application and the protection key. See also: <a href="#">"License Manager" on the next page</a>
<b>Key</b>	See <a href="#">"Sentinel protection keys" on page 381</a> .



<b>Key ID</b>	Unique identification number for a Sentinel protection key.
<b>License</b>	A logical entity (file or data) that enables the user to access a protected application (or part of it). The digital representation of a license is stored in a Sentinel protection key.
<b>License Manager</b>	<p>A component of Sentinel LDK that enables the protected application to locate and query the protection key that provides licensing authorization for the protected application to operate.</p> <p>The following types of License Managers exist: <a href="#">"Admin License Manager" on page 372</a>, <a href="#">"Integrated License Manager" on the previous page</a>, <a href="#">"External License Manager" on page 375</a></p>
<b>License Terms</b>	Detailed conditions and terms of usage contained in a license.
<b>License Type</b>	A set of license terms for a Feature. Each license model defines the conditions that control the use of a Feature in a Product.
<b>Locked Product</b>	A Product that is protected using Sentinel LDK and is locked to a specific machine or HL key. An <a href="#">"Unlocked Trialware Product" on page 382</a> becomes a Locked Product after the customer activates an entitlement for the Product.
<b>Locking Type</b>	Which types of protection keys can be used to license the Product. This determines the level of protection for a Product.
<b>Master key</b>	<p>A special-purpose Sentinel HL required for issuing licenses when the vendor works with Sentinel LDK-EMS on-premises or with Sentinel License Generation API. In these cases, the Master key contains the Sentinel LDK Master license. The Master key must be connected to the machine where Sentinel LDK-EMS or the License Generation API runs.</p> <p>For Sentinel EMS or Sentinel LDK-EMS hosted by Thales, Master key is not required. In this case, Thales recommends that you store the Master key in a secure location to prevent misuse.</p>
<b>Memory data</b>	Vendor-defined data (for example: passwords, values used by the software) that is specified in memory for a Product and transferred to the Sentinel protection key.
<b>Modification Product</b>	A modified version of an existing Product.

<b>On-demand Detach</b>	<p>You can manually detach a license from an SL key and attach it to your machine for a specified number of days. This is useful if you want to work with a protected application and expect to be disconnected from your company's network for an extended period.</p> <p>See also: <a href="#">"Detach" on page 375</a></p>
<b>Order</b>	<p>A request for a Product entitlement or protection key updates to be shipped to a customer.</p>
<b>Product</b>	<p>(Written with an uppercase "P") A licensing entity that represents one of a vendor's marketable software products or data files. The Product is coded into the memory of a Sentinel key and contains one or more Features. License terms are defined for each Feature in a Product.</p>
<b>Product Key</b>	<p>A string that is generated by Sentinel LDK-EMS and supplied to the end user for use as proof of purchase for Product Activation or Update Activation.</p>
<b>Production</b>	<p>The implementation of an order for Products or protection key updates.</p>
<b>Protect Once— Deliver Many— Evolve Often</b>	<p>The concept of separation between engineering and business processes, on which Sentinel LDK is designed.</p>
<b>Protection key</b>	<p>See <a href="#">"Sentinel protection keys" on page 381</a>.</p>
<b>Protection Key Memory</b>	<p>Secure memory that resides within a Sentinel protection key (HL or SL), for use by the protected software. Protection Key memory can be accessed or modified using the Sentinel Licensing API. The memory can be initialized when the key is generated, using data entered when defining the Product or when entering an order for a Product.</p>
<b>Protection Key Update</b>	<p>File containing update information for deployed Sentinel protection keys.</p> <p>See also: <a href="#">"V2C file" on page 382</a></p>
<b>Provisional Product</b>	<p>See <a href="#">"Unlocked Trialware Product" on page 382</a>.</p>
<b>R2H file</b>	<p>Recipient-to-Host file. This file is used to re-attach a cancelled detachable license to the host machine.</p>

<b>Real-time Clock (RTC)</b>	<p>A battery-powered clock that is available in the Sentinel HL Time key and Sentinel HL NetTime key. This clock is independent of the clock in the machine where the key is attached.</p> <p>See also: <a href="#">"V-Clock (Virtual Clock)" on page 382</a></p>
<b>Recipient machine</b>	<p>Remote machine to which a license that has been detached from a network pool on a host machine is temporarily attached.</p>
<b>Rehost</b>	<p>Transfer a Sentinel SL key from one end user computer to another. The rehost process is performed entirely by the end user, with no interaction with the vendor.</p>
<b>Reverse Engineering</b>	<p>Software attacks that are intended to unravel the algorithms and execution flow of a target program by tracing the compiled program to its source code. Sentinel Envelope protection implements contingency measures to repel such attacks and prevent crackers from discovering algorithms used inside protected software.</p>
<b>RUS utility</b>	<p>Sentinel Remote Update System (referred to as <i>RUS</i>) is an executable utility that the vendor can send to their end users to enable secure, remote updating of the license and memory data of Sentinel protection keys after they are deployed. See <a href="#">"Sentinel Remote Update System (RUS)" on page 381</a>.</p>
<b>RUS Generator</b>	<p>Tool that generates a RUS utility executable that is associated with the vendor's Batch Code and that is optionally branded and customized with additional text.</p>
<b>Script Envelope</b>	<p>Standalone tool for applying Envelope protection to Python applications. See also <a href="#">"Sentinel LDK Envelope" on the next page</a>.</p>
<b>Secure Storage</b>	<p>Area reserved by Sentinel LDK on a computer's local hard drive when one or more Sentinel SL protection keys are installed on the computer. The keys are installed in the secure storage area. This area can only be accessed or modified by Sentinel LDK components.</p>
<b>Secure Storage ID</b>	<p>A globally unique identifier of Secure storage on every machine.</p>
<b>Sentinel Admin API</b>	<p>API that enables administration of License Managers and Sentinel protection keys. Provides all the functionality that is available in Admin Control Center.</p> <p>See also: <a href="#">"License Manager" on page 377</a></p>
<b>Sentinel Admin Control Center</b>	<p>Customizable, Web-based, end-user utility that enables centralized administration of Admin License Managers and Sentinel protection keys.</p> <p>See also: <a href="#">"License Manager" on page 377</a>, <a href="#">"Admin License Manager" on page 372</a></p>

<b>Sentinel LDK-EMS</b>	Role-based application used to generate licenses and lock them to Sentinel protection keys, write specific data to the memory of a Sentinel protection key, and update licenses already deployed in the field. Sentinel LDK-EMS is installed as a service ( <i>Sentinel LDK-EMS Service</i> ) under Windows.
<b>Sentinel LDK-EMS Server</b>	Computer on which Sentinel LDK-EMS is installed and the Sentinel LDK-EMS Service is active.
<b>Sentinel HL key</b>	The hardware-based protection and licensing component of Sentinel LDK. One of the Sentinel protection key types.
<b>Sentinel HL Basic key</b>	Standard Sentinel HL local key that is used to protect software, and has a perpetual license. This key: <ul style="list-style-type: none"> <li>&gt; does not have any memory functionality.</li> <li>&gt; does not support concurrency or remote desktops.</li> <li>&gt; does not support V-Clock.</li> </ul>
<b>Sentinel HL network key</b>	Any Sentinel HL key that supports concurrency. This includes the following keys: <ul style="list-style-type: none"> <li>&gt; Sentinel HL Net key</li> <li>&gt; Sentinel HL NetTime key</li> <li>&gt; Any Sentinel HL (Driverless configuration) key except for Sentinel HL Basic keys</li> </ul>
<b>Sentinel HL (Driverless configuration) key</b>	Type of Sentinel HL key that does not require the Run-time Environment in order to protect an application or data file on a Windows machine.
<b>Sentinel HL (HASP configuration) key</b>	Type of Sentinel HL key that is fully compatible with protected applications that require the older HASP HL keys.
<b>Sentinel LDK-EMS (Thales Hosted) or LDKaaS</b>	Sentinel LDK-EMS hosted and managed by Thales. (This can be used instead of a local, on-premises Sentinel LDK-EMS installation.)
<b>Sentinel LDK - Demo Kit</b>	Kit containing software, hardware and documentation for evaluating the Sentinel LDK system.
<b>Sentinel LDK Envelope</b>	Application that wraps an application in a protective shield, ensuring that the protected application cannot run unless a specified Sentinel protection key is accessible by the program.

<b>Sentinel LDK Run time Environment (RTE)</b>	System component that enables communication between a Sentinel protection key and a protected application or data file. The Run-time Environment also contains Sentinel Admin Control Center.
<b>Sentinel LDK ToolBox</b>	GUI application designed to facilitate software engineers' use of various Sentinel LDK APIs and to generate source code.
<b>Sentinel License Manager</b>	See <a href="#">"License Manager" on page 377</a> .
<b>Sentinel Licensing API</b>	Interface for inserting calls to a Sentinel protection key
<b>Sentinel LDK Master license</b>	The license issued to you by Thales to work with Sentinel LDK. The Sentinel LDK Master license contain the modules that you purchased or subscribed to, as well as other license components. The Master license resides in your instance of Sentinel LDK-EMS hosted by Thales. If you installed Sentinel LDK-EMS on premises, your Master license resides in your Master key.
<b>Sentinel protection keys</b>	Sentinel HL keys and Sentinel SL keys.
<b>Sentinel Remote Update System (RUS)</b>	Utility that enables licenses in deployed Sentinel protection keys to be securely, remotely updated, or the contents of the keys to be modified. See also: <a href="#">"C2V file" on page 373</a> , <a href="#">"V2C file" on the next page</a>
<b>Sentinel SL key</b>	The software-based protection and licensing component of Sentinel LDK—a virtual Sentinel HL key.
<b>Developer keys</b>	The Master key and Developer key that contain your unique and private Vendor Codes. These keys enable you to apply protection to your programs, to program the Sentinel protection keys that you send to your end users, and to specify the license terms under which your software can be used.
<b>Status code</b>	Error or status message returned by the Thales system.
<b>Trialware</b>	Software or data files that can be distributed with an integrated Sentinel protection key for end-user evaluation during a limited time period. See also: <a href="#">"Unlocked Trialware Product" on the next page</a>

<b>Unlocked license</b>	<p>A license that does not lock a protected entity (application or data file) to a specific machine and does not necessarily impose any licensing restrictions on the use of the protected entity. The protected entity can be installed on any number of machines.</p> <p>The vendor creates an unlocked product from a base product. The vendor can use Sentinel LDK to protect the entity, and can either use a different mechanism to license the entity or can impose no license restrictions on the entity.</p>
<b>Unlocked Product</b>	A Product that is distributed with an <a href="#">"Unlocked license" above</a> . Unlocked Products are created from Base Products.
<b>Unlocked Trialware Product</b>	A Product that can be distributed as trialware, or during a grace period. Unlocked trialware Products are not locked to a specific machine and do not require activation for a limited period. Unlocked trialware Products typically have a duration of 30 to 90 days or 30 executions. This period can be set to begin either from the date of first use of the application or from the date that the license was generated. (The Unlocked trialware Product was formerly referred to as a <i>provisional Product</i> .)
<b>Unlocked Unlimited Product</b>	A Product that does not lock a protected application to a specific machine and does not necessarily impose any licensing restrictions on the use of the protected application. The Product can be granted a perpetual license or can be limited to any length of time that you choose. This enables the vendor to use Sentinel LDK to protect the application, but use a different mechanism to license the application (or impose no license restrictions on the application).
<b>UTC</b>	Coordinated Universal Time—the standard time common to every place in the world.
<b>V-Clock (Virtual Clock)</b>	<p>Virtual clock that is available in Sentinel SL keys and Sentinel HL (Driverless configuration) keys.</p> <p>See also: <a href="#">"Real-time Clock (RTC)" on page 379</a></p>
<b>V2C file</b>	<p><b>Vendor-to-Customer file.</b> This file is sent by the vendor to a customer. This file is generated either by Sentinel LDK-EMS or by other Sentinel LDK vendor tools. The file contains data to create or update a Sentinel protection key on the end user's computer. This data can include detailed changes to the license terms or data to be stored in the end users' Sentinel protection keys. V2C files can be safely sent using regular email. The naming convention for V2C files can be modified in Sentinel LDK-EMS.</p> <p>See also: <a href="#">"C2V file" on page 373</a></p>

<b>V2CP file</b>	<p><b>Vendor-to-Customer package file.</b> This file is generated only by Sentinel LDK-EMS. This file contains one or more V2C updates to a Sentinel protection key on the end user's computer. A V2CP file contains multiple V2C updates if Sentinel LDK-EMS determines that V2C transactions are pending at the time that it issues a new V2C transaction. The License Manager breaks down a V2CP file to its component V2C files and then applies each update in sequence. V2CP files can be safely sent using regular email.</p> <p>See also: <a href="#">"V2C file" on the previous page</a></p>
<b>Vendor Code</b>	A unique vendor-specific code that enables access to the vendor's Sentinel protection keys.
<b>Vendor ID</b>	A unique number that is associated with a given Vendor Code and Batch Code.
<b>Vendor libraries (Vlib)</b>	Vendor-specific API libraries. These libraries are built and customized on Thales servers. In this process, the libraries are customized differently for every vendor. These libraries are downloaded when you introduce one of your Vendor keys.
<b>Vendor keys</b>	Collective term used to refer to the Master key and the Developer key.