

SentinelHASP®

Sentinel HASP – v.5.10
Software Protection and Licensing Guide



Copyrights and Trademarks

Copyright © 2011 SafeNet, Inc. All rights reserved.

Cross-Locking, Hardlock, Hasp, HASP4, Method-Level Protection, Sentinel, Sentinel HASP, Sentinel HASP HL, Sentinel HASP SL, Sentinel HASP Business Studio, Sentinel HASP Reporting Module, Sentinel HASP Trialware, Sentinel SuperPro, and Sentinel UltraPro are either registered in United States Patent and Trademark Office or are trademarks of SafeNet, Inc. and its subsidiaries in the United States and/or other countries, and may not be used without written permission.

All other trademarks are property of their respective owners.

Patents

HASP® hardware and/or software products described in this document are protected by one or more of the following Patents, and may be protected by other United States and/or foreign patents, or pending patent applications: US 5,359,495, US 5,898,777, US 6,189,097, US 6,073,256, US 6,272,636, US 6,009,525, US 6,044,469, US 6,055,503, US 6,334,213, US 6,434,532, US 6,285,985, US 6,334,214, US 6,009,401, US 6,243,692, US 6,363,356, US 7,149,928, US 7,065,652, US 6,915,425, US 6,898,555, US 7,065,650, US 7,225,336, US 7,191,325, EP 1220075, EP 1318451, EP 1271310, EP 1353259, EP 1387235 and EP 1439446.

Disclaimer

We have attempted to make this document complete, accurate, and useful, but we cannot guarantee it to be perfect. When we discover errors or omissions, or they are brought to our attention, we endeavor to correct them in succeeding releases of the product. SafeNet, Inc., is not responsible for any direct or indirect damages or loss of business resulting from inaccuracies or omissions contained herein. The specifications contained in this document are subject to change without notice.

May 2011

Revision 0511-1-1

SAFENET SENTINEL HASP PRODUCT END USER LICENSE AGREEMENT

IMPORTANT INFORMATION - PLEASE READ THIS AGREEMENT CAREFULLY BEFORE USING THE CONTENTS OF THE PACKAGE AND/OR BEFORE DOWNLOADING OR INSTALLING THE SOFTWARE PRODUCT. ALL ORDERS FOR AND USE OF THE SENTINEL HASP® PRODUCTS (including without limitation, the Developer's Kit, libraries, utilities, diskettes, CD_ROM, DVD, Sentinel HASP keys, the software component of SafeNet Sentinel HASP™ and the Sentinel HASP Software Protection and Licensing Guide) (hereinafter "**Product**") SUPPLIED BY **SAFENET, INC.**, (or any of its affiliates - either of them referred to as "**SAFENET**") ARE AND SHALL BE, SUBJECT TO THE TERMS AND CONDITIONS SET FORTH IN THIS AGREEMENT.

BY OPENING THE PACKAGE CONTAINING THE PRODUCTS AND/OR BY DOWNLOADING THE SOFTWARE (as defined hereunder) AND/OR BY INSTALLING THE SOFTWARE ON YOUR COMPUTER AND/OR BY USING THE PRODUCT, YOU ARE ACCEPTING THIS AGREEMENT AND AGREEING TO BE BOUND BY ITS TERMS AND CONDITIONS.

IF YOU DO NOT AGREE TO THIS AGREEMENT OR ARE NOT WILLING TO BE BOUND BY IT, DO NOT OPEN THE PACKAGE AND/OR DOWNLOAD AND/OR INSTALL THE SOFTWARE AND PROMPTLY (at least within 7 days from the date you received this package) RETURN THE PRODUCTS TO SAFENET, ERASE THE SOFTWARE, AND ANY PART THEREOF, FROM YOUR COMPUTER AND DO NOT USE IT IN ANY MANNER WHATSOEVER.

This Agreement has 3 sections:

Section I applies if you are downloading or using the Product free of charge for evaluation purposes only.

Section II applies if you have purchased or have been otherwise granted by SafeNet a license to use the Product.

Section III applies to all grants of license.

1. **SECTION I** - TERMS APPLICABLE TO GRANT OF EVALUATION LICENSE

- 1.1 **License Grant.** SafeNet hereby grants to you, and you accept, a nonexclusive license to use the Product in machine-readable, object code form only, free of charge, for the purpose of evaluating whether to purchase an ongoing license to the Product and only as authorized in this License Agreement. The evaluation period is limited to the maximum amount of days specified in your applicable evaluation package. You may use the Product, during the evaluation period, in the manner described in Section III below under “Extent of Grant.”
- 1.2 **DISCLAIMER OF WARRANTY.** The Product is provided on an “AS IS” basis, without warranty of any kind. IMPLIED WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE, SATISFACTION AND MERCHANTABILITY SHALL NOT APPLY. SOME JURISDICTIONS DO NOT ALLOW EXCLUSIONS OF AN IMPLIED WARRANTY, SO THIS DISCLAIMER MAY NOT APPLY TO YOU AND YOU MAY HAVE OTHER LEGAL RIGHTS THAT VARY BY JURISDICTION. The entire risk as to the quality and performance of the Product is borne by you. This disclaimer of warranty constitutes an essential part of the agreement.

If you initially acquired a copy of the Product without purchasing a license and you wish to purchase a license, contact SafeNet or any SafeNet representative.

2. **SECTION II** - APPLICABLE TERMS WHEN GRANTED A LICENSE

- 2.1 **License Grant.** Subject to your payment of the license fees applicable to the type and amount of licenses purchased by you and set forth in your applicable purchase order, SafeNet hereby grants to you, and you accept, a personal, nonexclusive and fully revocable limited License to use the Software (as such term is defined in Section III hereunder, in the Intellectual Property subsection), in executable form only, as described in the Software accompanying user documentation and only according to the terms of this Agreement: (i) you may install the Software and use it on computers located in your place of business, as described in SafeNet’s related documentation; (ii) you may merge and link the Software into your computer programs for the sole purpose described in the Sentinel HASP Software Protection and Licensing Guide; however, any portion of the Software merged into another computer program shall be deemed as derivative work and will continue to be subject to the terms of this Agreement; and (iii) you are permitted to make a reasonable number of copies of the Software solely for backup purposes. The Software shall not be used for any other purposes.
- 2.2 **Sub-Licensing.** After merging the Software in your computer program(s) according to the License Grant section above, you may sub-license, pursuant to the terms of this Agreement, the merged Software and resell the hardware components of the Product, which you purchased from SafeNet, if applicable, to distributors and/or users. Preceding such a sale and sub-licensing, you shall make sure that your contracts with any of your distributors and/or end users (and their contracts with their customers) shall

contain warranties, disclaimers, limitation of liability, and license terms which are no less protective of SafeNet's rights than such equivalent provisions contained herein. In addition, you shall make it abundantly clear to your distributors and/or end users, that SafeNet is not and shall not, under any circumstances, be responsible or liable in any way for the software and software licenses contained in your computer programs which you merge with the SafeNet Software and distribute to your distributors and/or end users, including, without limitation, with respect to extending license terms and providing maintenance for any software elements and/or computer programs which are not the SafeNet Software. SafeNet expressly disclaims any responsibility and liability with respect to any computer programs, software elements, and/or hardware elements which are not and do not form part of the SafeNet product.

- 2.3 **Limited Warranty.** SafeNet warrants, for your benefit alone, that (i) the Software, when and as delivered to you, and for a period of three (3) months after the date of delivery to you, will perform in substantial compliance with the Sentinel HASP Software Protection and Licensing Guide, provided that it is used on the computer hardware and with the operating system for which it was designed; and (ii) that the Sentinel HASP key, for a period of twelve (12) months after the date of delivery to you, will be substantially free from significant defects in materials and workmanship. You may enable or disable certain features when applying the Sentinel HASP protection software by changing settings in the Sentinel HASP tools in accordance with the Sentinel HASP Software Protection and Licensing Guide; HOWEVER, IT IS IMPORTANT TO NOTE THAT WHEN ENABLING OR DISABLING SOME FEATURES YOU MIGHT REDUCE THE LEVEL OF PROTECTION PROVIDED BY THE SOFTWARE.
- 2.4 **Warranty Disclaimer.** SAFENET DOES NOT WARRANT THAT ANY OF ITS PRODUCT(S) WILL MEET YOUR REQUIREMENTS OR THAT THEIR OPERATION WILL BE UNINTERRUPTED OR ERROR-FREE. TO THE EXTENT ALLOWED BY LAW, SAFENET EXPRESSLY DISCLAIMS ALL EXPRESS WARRANTIES NOT STATED HERE AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NO SAFENET'S DEALER, DISTRIBUTOR, RESELLER, AGENT OR EMPLOYEE IS AUTHORIZED TO MAKE ANY MODIFICATIONS, EXTENSIONS, OR ADDITIONS TO THIS WARRANTY. If any modifications are made to the Software or to any other part of the Product by you; if the media and the Sentinel HASP key is subjected to accident, abuse, or improper use; or if you violate any of the terms of this Agreement, then the warranty in Section 2.3 above, shall immediately be terminated. The warranty shall not apply if the Software is used on or in conjunction with hardware or program other than the unmodified version of hardware and program with which the Software was designed to be used as described in the Sentinel HASP Software Protection and Licensing Guide.
- 2.5 **Limitation of Remedies.** In the event of a breach of the warranty set forth above, SafeNet's sole obligation, and your sole remedy shall be, at SafeNet's sole discretion: (i) to replace or repair the Product, or component thereof, that does not meet the foregoing limited warranty, free of charge; or (ii) to refund the price paid by you for the Product, or component thereof. Any replacement or repaired component will be warranted for the remainder of the original warranty period or 30 days, whichever is longer. Warranty claims must be made in writing during the warranty period and within seven (7) days of the observation of the defect accompanied by evidence satisfactory

to SafeNet. All Products should be returned to the distributor from which they were purchased (if not purchased directly from SafeNet) and shall be shipped by the returning party with freight and insurance paid. The Product or component thereof must be returned with a copy of your receipt.

3. **SECTION III - TERMS APPLICABLE TO ALL GRANTS OF LICENSE**

- 3.1 **Extent of Grant and Prohibited Uses**. Except as specifically permitted in Sections 2.1 and 2.2 above, you agree not to (i) use the Product in any manner beyond the scope of license purchased by you in accordance with your applicable purchase order; (ii) use, modify, merge or sub-license the Software or any other of SafeNet's products except as expressly authorized in this Agreement and in the Sentinel HASP Software Protection and Licensing Guide; and (iii) sell, license (or sub-license), lease, assign, transfer, pledge, or share your rights under this License with/to anyone else; and (iv) modify, disassemble, decompile, reverse engineer, revise or enhance the Software or attempt to discover the Software's source code; and (v) place the Software onto a server so that it is accessible via a public network; and (vi) use any back-up or archival copies of the Software (or allow someone else to use such copies) for any purpose other than to replace an original copy if it is destroyed or becomes defective. If you are a member of the European Union, this agreement does not affect your rights under any legislation implementing the EC Council Directive on the Legal Protection of Computer Programs. If you seek any information within the meaning of that Directive you should initially approach SafeNet.
- 3.2 **Intellectual Property**. THIS IS A LICENSE AGREEMENT AND NOT AN AGREEMENT FOR SALE. The software component of the SafeNet Sentinel HASP Product, including any revisions, corrections, modifications, enhancements, updates and/or upgrades thereto, (hereinafter in whole or any part thereof defined as: "**Software**"), and the related documentation, ARE NOT FOR SALE and are and shall remain in SafeNet's sole property. All intellectual property rights (including, without limitation, copyrights, patents, trade secrets, trademarks, etc.) evidenced by or embodied in and/or attached/connected/related to the Product, (including, without limitation, the Software code and the work product performed in accordance with Section II above) are and shall be owned solely by SafeNet. This License Agreement does not convey to you an interest in or to the Software but only a limited right of use revocable in accordance with the terms of this License Agreement. Nothing in this Agreement constitutes a waiver of SafeNet's intellectual property rights under any law.
- 3.3 **Audit**. SafeNet shall have the right, at its own expense, upon reasonable prior notice, to periodically inspect and audit your records to ensure your compliance with the terms and conditions of this license agreement.
- 3.4 **Termination**. Without prejudice to any other rights, SafeNet may terminate this license upon the breach by you of any term hereof. Upon such termination by SafeNet, you agree to destroy, or return to SafeNet, the Product and the Documentation and all copies and portions thereof.

- 3.5 **Limitation of Liability.** SafeNet's cumulative liability to you or any other party for any loss or damages resulting from any claims, demands, or actions arising out of or relating to this Agreement and/or the use of the Product shall not exceed the license fee paid to SafeNet for the use of the Product/s that gave rise to the action or claim, and if no such Product/s is/are so applicable then SafeNet's liability shall not exceed the amount of license fees paid by You to SafeNet hereunder during the twelve (12) months period preceding the event. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, TORT, CONTRACT, OR OTHERWISE, SHALL SAFENET OR ITS SUPPLIERS OR RESELLERS OR AGENTS BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY TYPE INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, BUSINESS INTERRUPTION, COMPUTER FAILURE OR MALFUNCTION, LOSS OF BUSINESS PROFITS, LOSS OF BUSINESS INFORMATION, DAMAGES FOR PERSONAL INJURY OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SAFENET SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION AND EXCLUSION MAY NOT APPLY TO YOU.
- 3.6 **No other Warranties.** Except and to the extent specifically provided herein, SafeNet makes no warranty or representation, either express or implied, with respect to its Products as, including their quality, performance, merchantability or fitness for a particular purpose.
- 3.7 **Export Controls.** YOU ACKNOWLEDGE THAT THE SOFTWARE IS SUBJECT TO REGULATION BY UNITED STATES, EUROPEAN UNION, AND/OR OTHER GOVERNMENT AGENCIES, WHICH PROHIBIT EXPORT OR DIVERSION OF THE SOFTWARE TO CERTAIN COUNTRIES AND CERTAIN PERSONS. YOU AGREE TO COMPLY WITH ALL EXPORT LAWS, REGULATIONS AND RESTRICTIONS OF THE UNITED STATES DEPARTMENT OF STATE, DEPARTMENT OF COMMERCE OR OTHER LEGAL AUTHORITY WITHIN THE UNITED STATES OR ANY FOREIGN ENTITY WHICH REGULATES THEIR SHIPMENT. YOU WILL NOT EXPORT IN ANY MANNER, EITHER DIRECTLY OR INDIRECTLY, ANY SOFTWARE OR ANY PRODUCT THAT INCORPORATES ANY SOFTWARE WITHOUT FIRST OBTAINING ALL NECESSARY APPROVAL FROM APPROPRIATE GOVERNMENT AGENCIES. YOU AGREE TO INDEMNIFY SAFENET AGAINST ALL CLAIMS, LOSSES, DAMAGES, LIABILITIES, COSTS AND EXPENSES, INCLUDING REASONABLE ATTORNEYS' FEES, TO THE EXTENT SUCH CLAIMS ARISE OUT OF ANY BREACH OF THIS SECTION 3.7.
- 3.8 **Governing Law & Jurisdiction.** This License Agreement shall be construed, interpreted and governed by the laws of the State of Delaware without regard to conflicts of laws and provisions thereof. The exclusive forum for any disputes arising out of or relating to this License Agreement shall be an appropriate federal or state court sitting in Harford County,

State of Maryland, USA. The application of the United Nations Convention of Contracts for the International Sale of Goods is expressly excluded. The failure of either party to enforce any rights granted hereunder or to take action against the other party in the event of any breach hereunder shall not be deemed a waiver by that party as to subsequent enforcement of rights or subsequent actions in the event of future breaches.

- 3.9 **Third Party Software.** The Product contains the **Open SSL Toolkit** which includes the OpenSSL software, as set forth in **Exhibit A** and the Original SSLeay software, as set forth in **Exhibit B**. Such third party's software is provided "As Is" and use of such software shall be governed by the terms and conditions as set forth in **Exhibit A** and **Exhibit B**. If the Product contains any software provided by third parties other than the software noted in **Exhibit A** and **Exhibit B**, such third party's software are provided "As Is" and shall be subject to the terms of the provisions and condition set forth in the agreements contained/attached to such software. In the event such agreements are not available, such third party's software are provided "As Is" without any warranty of any kind and this Agreement shall apply to all such third party software providers and third party software as if they were SafeNet and the Product respectively.
- 3.10 **Miscellaneous.** If the copy of the Product you received was accompanied by a printed or other form of "hard-copy" End User License Agreement whose terms vary from this Agreement, then the hard-copy End User License Agreement governs your use of the Product. This Agreement represents the complete agreement concerning this license and may be amended only by a writing executed by both parties. THE ACCEPTANCE OF ANY PURCHASE ORDER PLACED BY YOU, IS EXPRESSLY MADE CONDITIONAL ON YOUR ASSENT TO THE TERMS SET FORTH HEREIN, COMBINED WITH THE APPLICABLE LICENSE SCOPE AND TERMS, IF ANY, SET FORTH IN YOUR PURCHASE ORDER. If any provision of this Agreement is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. The failure of either party to enforce any rights granted hereunder or to take action against the other party in the event of any breach hereunder shall not be deemed a waiver by that party as to subsequent enforcement of rights or subsequent actions in the event of future breaches.

Exhibit A

Open SSL License

A. Notices

- I. Copyright (c) 1998-2005 The OpenSSL Project. All rights reserved.
- II. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- III. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- IV. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"
- V. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
- VI. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
- VII. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"

B. DISCLAIMER OF WARRANTY

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Exhibit B

Original SSLeay License

A. Notices

- I. Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com). All rights reserved.
- II. This package is an SSL implementation written by Eric Young (eay@cryptsoft.com).
- III. The implementation was written so as to conform with Netscapes SSL.
- IV. This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).
- V. Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed.
- VI. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used.
- VII. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.
- VIII. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- IX. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- X. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)".
- XI. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

B. DISCLAIMER OF WARRANTY.

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Certifications

CE Compliance



The HASP product line complies with the CE EMC Directive and related standards*. HASP products are marked with the CE logo and a HASP CE conformity card is included in every shipment or upon demand.

*EMC directive 89/336/EEC and related standards EN 55022, EN 50082-1.

FCC Compliance

FCC authorities have determined that HASP is not a Class B Computing Device Peripheral and therefore does not require FCC regulation.

UL Certification

The HASP product line successfully completed UL 94 Tests for Flammability of Plastic Materials for Parts in Devices and Appliances. HASP products comply with UL 1950 Safety of Information Technology Equipment regulations.

ISO 9001:2000 Certification



The HASP product line is designed and manufactured by Aladdin Knowledge Systems, Inc., an ISO 9001:2000 certified company. Aladdin's quality assurance system is approved by the International Organization for Standardization (ISO), ensuring that Aladdin products and customer service standards consistently meet specifications in order to provide outstanding customer satisfaction.

Certificate of Compliance

Upon request, SafeNet Inc./Aladdin Knowledge Systems, Inc. will supply a Certificate of Compliance to any software developer who wishes to demonstrate that the HASP product line conforms to the specifications stated. Software developers can distribute this certificate to the end user along with their programs.

Contents

Familiarizing Yourself with Sentinel HASP Vendor Suite	23
Contents of Your Sentinel HASP Kit	23
Sentinel HASP Developer Kit.....	23
Sentinel HASP Starter Kit	24
About This Guide.....	24
Obtaining Support.....	26
Training.....	26
Technical Support.....	26

Part 1 Getting Started

1. Understanding Sentinel HASP Software Protection and Licensing	29
Fundamentals of Protection	30
What is Protection?	30
Major Protection Solutions	31
Hardware-based Solutions.....	31
Software-based Solutions	31
Comparative Benefits of Hardware-based and Software-based Solutions.....	32
Advantages of a Combined Solution	32
Fundamentals of Licensing	33
Flexible and Secure Licensing Solutions	34
Licensing Planning and Models	34
Updating and Enforcing Usage Terms	35
Principles of Sentinel HASP	36
Protect Once—Deliver Many	36
Cross-locking.....	37
Mixing and Matching Licenses and Sentinel HASP Protection Keys.....	37

Customizing Your Unique Solution	38
Personalized Vendor and Batch Codes	39
Selecting the Best Key for Your Requirements	39
Sentinel HASP Vendor Keys	40
End-User Keys	40
Sentinel HASP Protection Process	45
Obtaining Additional Information About Sentinel HASP	46

Part 2 Protection

2. Protecting Software	49
Sentinel HASP Protection	49
Elements of Sentinel HASP Protection	50
Selecting a Protection Method	53
3. Sentinel HASP Run-time API Protection.....	57
Overview	57
Universal Sentinel HASP Run-time API	58
Sentinel HASP Run-time API Prerequisites	59
Learning the Sentinel HASP Run-time API	62
Sentinel HASP ToolBox	62
Sentinel HASP Run-time API Samples	63
Implementation	64
Planning Your Requirements	64
Sentinel HASP Run-time API Workflow	65
Sentinel HASP Run-time API Login Function	66
Sentinel HASP Run-time API Functionality	68
Function Groups	68
4. Sentinel HASP Envelope Protection	71
Functionality	71
Basic Protection Workflow	73
Required and Optional Protection Parameters	76
General Customizable Parameters	76
Sentinel HASP Envelope for Windows	78
Prerequisites for Windows	78

Running Sentinel HASP Envelope.....	78
Protecting Win32 or Windows x64 Programs.....	80
Accessing and Encrypting Data Files for Windows Programs	81
Running Sentinel HASP Envelope from a Windows Command-line	84
Protecting .NET Assemblies.....	85
.NET Considerations	86
Global Features in .NET Assemblies.....	86
Method-level Protection	87
Code and Symbol Obfuscation in .NET Assemblies	90
Defining Sentinel HASP Envelope Protection Settings in Source Code	91
Sentinel HASP Envelope for Linux Applications.....	93
Sentinel HASP Envelope Prerequisites for Linux.....	93
Protecting Linux Applications.....	94
Sentinel HASP Envelope for Mac Binaries	96
Sentinel HASP Envelope Prerequisites for Mac.....	96
Running Sentinel HASP Envelope for Mac	96
Sentinel HASP Envelope for Mac Protection Parameters.....	97
Sentinel HASP Envelope for Java Executables.....	97
Java Considerations.....	98
Sentinel HASP Envelope Prerequisites for Java	98
Running Sentinel HASP Envelope for Java Engines	99
Sentinel HASP Envelope for Java Protection Parameters.....	99
Protecting Java Executables.....	100
5. Protection Strategies	101
Overview	101
General Protection Guidelines	102
Types of Attack and Their Sentinel HASP Defense.....	104
6. Working with the DataHASP Encryption Utility	109
Introduction	109
When to Encrypt Data Files	110
DataHASP Users	110
Sentinel HASP Data File Handling	110

DataHASP Prerequisites	111
Launching DataHASP	112
Supported Functionality	112
Modifying Input	113

Part 3 Licensing

7. Introduction to Sentinel HASP Business Studio	117
Sentinel HASP Business Studio Overview	118
Sentinel HASP Business Studio Major Workflows	118
Sentinel HASP Business Studio Users and User Roles	121
Distribution Channels	123
Getting Started With Sentinel HASP Business Studio	124
Prerequisites for the Sentinel HASP Administrator	125
Sentinel HASP Business Studio Window	126
Using the Sentinel HASP Business Studio Help	127
8. Preparing Your Sentinel HASP Licensing Plan	129
Licensing Overview	130
Preparing Your Licensing Plan	131
Identifying Functional Components (Features)	132
Combining Features Into Products	133
Choosing the Protection Level for Your Products	134
HASP HL Key Protection and Activation	135
HASP SL Key Protection and Activation	135
Specifying the Protection Level for Individual Orders	136
Designating Products for Trial or Grace Period Use	137
Assigning License Terms to Features	138
Specifying License Values for Individual Orders	138
Utilizing HASP Memory	140
Using Your Licensing Plan With Sentinel HASP Business Studio	140
9. Implementing Your Sentinel HASP Licensing Plan	141
License Planning in Sentinel HASP Business Studio	142
Managing Features	143

Defining Features	143
Withdrawing a Feature	145
Managing Products.....	145
Defining New Products.....	147
Defining Provisional Products	155
Product Status Values	155
Duplicating a Product.....	156
Withdrawing a Product.....	156
Maintaining Products and Licenses	156
Managing Product Versions	157
Canceling Product Licenses	159
10. Sentinel HASP Orders, Production, and Development Tasks	163
Sentinel HASP Order Processing and Production.....	164
Managing Orders.....	165
Defining Orders	166
Order Status Values	171
Processing C2V Information	172
Order Processing and Production Examples.....	173
Producing Orders	176
Producing HASP HL Key Orders	177
Producing Orders for Product Keys	178
Producing HASP Update Orders	178
Performing Development-related Tasks.....	179
Generating Bundles of Provisional Products	179
Generating the Sentinel HASP Run-time Environment Installer	181
Exporting Definition Data	182
Customizing and Branding RUS	182
Enabling Trial Use and Grace Periods.....	183
Example 1: Issuing a Provisional Product for Trial Use.....	183
Example 2: Issuing a Product for a Grace Period	184
11. Sentinel HASP Administration and Customer Services	185
Administration Tasks.....	186
Maintaining User Details	187

Maintaining Sentinel HASP Master Keys	188
Configuring System Settings	190
Customer Services.....	191
12. Sentinel HASP Remote Update System.....	193
RUS Overview	193
RUS Workflow.....	194
Using RUS	196
Instructions for Customers Using Sentinel HASP RUS	196
13. Generating Sentinel HASP Reports.....	199
Reports Facility Overview	200
Permissions for Working With Reports.....	200
Scheduling Reports.....	201
Presentation Formats.....	201
Export Formats	202
Available Reports	203

Part 4 Distributing Sentinel HASP Software

14. Distributing Sentinel HASP With Your Software	207
Sentinel HASP Software for End Users	207
Protection-related Software.....	208
Network Environment Management.....	209
Software for Updating Licenses	209
Distributing Sentinel HASP Run-time Environment.....	209
Sentinel HASP Run-time Environment for Windows	209
Sentinel HASP Run-time Environment for Mac	214
Sentinel HASP Run-time Environment for Linux	215
15. Sentinel HASP Admin Control Center.....	217
Introduction to Admin Control Center	217
Launching Admin Control Center	218
Admin Control Center Interface	219
Administrator's Workflow	220
Configuration Considerations	221

Configuring Detachable License Definitions	223
Configuring Log File Output	224
Diagnostics	224
Applying Basic Configuration Changes Globally	225
Customizing Admin Control Center Look and Feel	225
Writing Templates	226
Configuring Admin Control Center to Use Your Custom Template	230

Part 5 Sentinel HASP Licensing Models

16. Sentinel HASP Licensing Models: Overview	235
Introduction	235
Sentinel HASP Licensing	236
17. Sentinel HASP Licensing Models: Description of Models.....	241
Evaluation Licensing Models	241
Trialware	242
High-security Time-limited Evaluation	243
Execution-limited Evaluation	244
Demoware	246
Component-based Licensing Models	247
Module-based (Suites)	248
Feature-based	249
Metered Licensing Models	250
Time-limited Rental	251
Phased Rental	252
Micro-rental	253
Subscription	255
Pay-by-Peak Time (Peak Time)	256
Time-based Overdraft	258
Standard Counter	259
Phased Counter	260
Capacity (CPU/Memory/Disk)	262
Locked Licenses	263
Machine-locked	263

User-locked.....	265
Mobile Licenses	267
Portable.....	267
Commuter	268
Software on a Key	269
Network Licensing Models.....	269
Limited Concurrent End Users in a Network.....	270
Time-limited Concurrent End Users in a Network.....	271
Execution-limited Concurrent End Users in a Network.....	273
Volume	274
Site	276
Sales Boosting Licensing Models.....	277
KickStart (Quick-delivery Grace).....	277
Referral-based Sales.....	280
Automatic Sales Agent	281
Reselling Via Distribution Channels	283
Perpetual Licensing model	285

Part 6 Appendices

A. Troubleshooting	289
Checklist.....	290
Problems and Solutions.....	290
B. Sentinel HASP Run-time API Reference.....	293
Introduction	293
API Function Overview	294
Detailed Description of Functions	296
hasp_datetime_to_hasptime().....	297
hasp_decrypt().....	299
hasp_detach().....	301
hasp_encrypt().....	304
hasp_free()	306
hasp_get_rtc().....	307
hasp_get_sessioninfo()	308

hasp_get_info()	310
hasp_get_size()	312
hasp_hasptime_to_datetime()	314
hasp_login()	316
hasp_login_scope()	319
hasp_logout()	321
hasp_read()	322
hasp_update()	324
hasp_write()	326
API Status Codes	328
Blinking the HASP HL Key LED	331
Overview	331
How to Blink the HASP HL Key	332
C. Sentinel HASP Run-time Network Activity	333
Local Communications	334
Remote Communications	335
D. How Sentinel HASP Detects Machine Cloning	337
Overview	337
Clone Detection for a Physical Machine	338
Clone Detection for a Virtual Machine	339
E. How Sentinel HASP Protects Time-based Licenses Locked to HASP SL Keys	343
F. Understanding the Sentinel HASP Master Key Licenses	345
Licensing Concepts	346
Activation Module License	347
Activations Pool	347
Network Seats Pool	349
Provisional Products License	352
Reporting Module License	353
G. Glossary	355
Index	361

Familiarizing Yourself with Sentinel HASP Vendor Suite

This front matter consolidates various snippets of information related to Sentinel HASP Vendor Suite. It is recommended that you review this information to familiarize yourself with:

- The contents of your Sentinel HASP Starter or Developer kit
- The information provided in this Guide
- How to obtain additional technical support for these products

Contents of Your Sentinel HASP Kit

Two Sentinel HASP kits are available as part of the Sentinel HASP Vendor Suite—the Sentinel HASP Developer Kit enables you to evaluate Sentinel HASP protection and licensing; the Sentinel HASP Starter Kit enables you to apply Sentinel HASP protection and licensing to your software.

Sentinel HASP Developer Kit

The Sentinel HASP Developer Kit contains the software and hardware you need to evaluate Sentinel HASP protection and licensing. The following items are included:

- Sentinel HASP Vendor Suite software on a single DVD
- HASP HL Demo keys to facilitate the evaluation process

- Sentinel HASP Software Protection and Licensing Quick Start card
- Sentinel HASP Software Protection and Licensing Guide (this book)
- Sentinel HASP Software Protection and Licensing Tutorial
- Sentinel HASP Business Studio Server Installation Guide

Sentinel HASP Starter Kit

The Sentinel HASP Starter Kit contains the software and hardware you need to apply Sentinel HASP protection and licensing. The following items are included:

- Sentinel HASP Vendor Suite software on a single DVD
- Sentinel HASP Vendor keys:
 - ◆ Sentinel HASP Developer key for applying protection
 - ◆ Sentinel HASP Master key for defining and applying license terms
- Sentinel HASP Software Protection and Licensing Quick Start card
- Sentinel HASP Software Protection and Licensing Guide (this book)
- Sentinel HASP Software Protection and Licensing Tutorial
- Sentinel HASP Business Studio Server Installation Guide
- HASP HL keys for distribution to your customers, according to your order

About This Guide

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

The guide is divided into the following parts.

■ **Part 1—Getting Started**

Introduces Sentinel HASP, presents basic protection and licensing concepts, and leads you through the process of configuring the system. You should read this part after opening your Sentinel HASP Developer's or Starter's kit.

■ **Part 2—Protection**

Provides an in-depth presentation of Sentinel HASP protection methods. This part includes strategies for maximizing the protection of your software using Sentinel HASP. This part is specifically for software engineers who have the responsibility for using the Sentinel HASP protection applications to protect software.

■ **Part 3—Licensing**

Discusses the options that Sentinel HASP 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 Sentinel HASP Software**

Details the Sentinel HASP software that can be delivered to end users to ensure optimal performance of protected software. This part also describes the various ways of effectively delivering the Sentinel HASP software components.

■ **Part 5—Sentinel HASP Licensing Models**

Provides an overview and detailed description of the various Sentinel HASP Licensing models that you can use to distribute your software.

■ **Part 6—Appendices**

Provides the following supplementary information:

- ◆ A troubleshooting section that identifies various issues that you may encounter and provides solutions
- ◆ A comprehensive glossary with concise explanations of Sentinel HASP terms
- ◆ The Sentinel HASP Run-time API reference, which provides a list of functions and their parameters, and their return values

Obtaining Support

SafeNet has both international offices and many local distributors providing support for Sentinel HASP—virtually whenever and wherever required. To find the name of your nearest office or distributor, go to the following URL:

<http://www.safenet-inc.com/Contact-Us.aspx>

Training

For additional information and training about Sentinel HASP implementation issues, contact our team of international consultants at the URL provided above. The consultants can provide you with tailored training sessions on the following:

- Integration of Sentinel HASP into your product
- Analysis of the best protection strategy for your applications
- Assistance in implementation of your protection and licensing models

Technical Support

You can download updates, executables, and documentation using the following URL:

<http://www.safenet-inc.com/technicalsupport.aspx>

You can contact our Technical Support team using the following URL:

<http://www.safenet-inc.com/Contact-Us.aspx>

Part 1 Getting Started

In this section:

- **Chapter 1: Understanding Sentinel HASP Software Protection and Licensing**
Provides an overview of the concepts of software and intellectual property protection and licensing, discusses the primary protection solutions, and focuses on how Sentinel HASP provides a comprehensive solution to all your protection requirements.

Chapter 1

Understanding Sentinel HASP Software Protection and Licensing

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

SafeNet recommends that you familiarize yourself with the information in this chapter so that you can maximize the benefits of using Sentinel HASP.

In this chapter:

- Fundamentals of Protection
- Major Protection Solutions
- Fundamentals of Licensing
- Flexible and Secure Licensing Solutions
- Principles of Sentinel HASP
- Customizing Your Unique Solution

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.

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

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 HASP provides a variety of hardware devices in the form of HASP HL keys. You can select the type of HASP HL key that best suits your requirements. For more information about HASP HL keys, see [HASP HL Keys](#) on page 41.

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

For more information about HASP SL keys, see [HASP SL Keys](#) on page 41.

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.

It is probable that you 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

It is likely that your strategies will also require the following:

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

Sentinel HASP Combined Solution

Sentinel HASP 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 license models according to the requirements of your market, and apply their usage terms independently of the protection process
- Select hardware-based (HASP HL) or software-based (HASP 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 fully-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 enforce your defined usage terms with secure licensing methods.

Flexible and Secure Licensing Solutions

Sentinel HASP 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 models required to meet the company's software distribution needs.

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

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

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

- Trialware (try-before-you-buy)
- Rental/Subscription
- Module-/Feature-based
- Floating Usage
- Time-based
- Execution-based

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

The separation of the engineering and licensing processes embodied in Sentinel HASP makes it possible to modify the 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 HASP 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 HASP technology prevents tampering with usage terms.

Principles of Sentinel HASP

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

- *Protect Once—Deliver Many*, which is the concept of separating the Sentinel HASP engineering and business processes.
- *Cross-Locking*, which is the technology that supports the *Protect Once—Deliver Many* concept, enabling a protected application to work with a HASP HL or a HASP SL key.

Protect Once—Deliver Many

At the heart of Sentinel HASP lies the *Protect Once—Deliver Many* concept. *Protect Once—Deliver Many* 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 HASP Envelope, Sentinel HASP ToolBox and the Sentinel HASP Run-time API protection tools.

The business processes—that is, software licensing and selection of the appropriate Sentinel HASP protection key—are performed by business management using Sentinel HASP Business Studio.

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

Cross-locking

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

The decision about the type of Sentinel HASP 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 HASP Protection Keys

Sentinel HASP gives you complete flexibility to choose the combination of license and Sentinel HASP 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 HASP SL key. Additionally, you may choose to ship the same protected software with a network license that is locked to a HASP HL key, and allow users unlimited access to all features.

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

Customizing Your Unique Solution

Sentinel HASP 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 HASP 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 HASP protection key, even though the selection of key is determined at a later time.
- *Sentinel HASP ToolBox* and the *Sentinel HASP Run-time API* enable you to enhance the protection offered by Sentinel HASP Envelope, by incorporating complex protection mechanisms into your source code.
- *Sentinel HASP Business Studio* enables you to create licenses and lock them to Sentinel HASP protection keys, to write specific data to the memory of a Sentinel HASP protection key, and to update licenses already deployed in the field. These processes are performed independently of the protection process.
- Customized Sentinel HASP Vendor keys are used in-house by your staff, together with Sentinel HASP state-of-the-art security applications.
- A selection of *Sentinel HASP protection keys* enable you to meet the specific requirements of your business. Your unique Sentinel HASP 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 HASP Vendor Suite.

Personalized Vendor and Batch Codes

When you purchase a Sentinel HASP Starter kit from SafeNet, you are provided with Sentinel HASP Vendor keys that contain unique Vendor Codes that are specific to your company. The codes are used by Sentinel HASP to communicate with your Sentinel HASP protection keys, and to differentiate your keys from those of other software vendors.

Vendor Code

The Vendor Code is a unique confidential code assigned to you by SafeNet when you place your first order for Sentinel HASP protection keys. It is integrated into your Sentinel HASP Vendor keys. When you are protecting your software and licenses to Sentinel HASP protection keys for distribution, the Vendor Code is extracted from your Sentinel HASP Vendor keys.

Batch Code

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

Selecting the Best Key for Your Requirements

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

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

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

A variety of Sentinel HASP 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 HASP Vendor Keys

When you purchase Sentinel HASP, you are provided with two Sentinel HASP Vendor keys—the Sentinel HASP Master key and the Sentinel HASP Developer key. These keys enable you to apply protection to your programs, program the Sentinel HASP protection keys that you send to your end users, and to specify the license terms under which your software can be used.

Sentinel HASP Developer Key

The Sentinel HASP Developer key is used by your programmers in conjunction with the Sentinel HASP protection tools to protect your software and data files.

Sentinel HASP Master Key

The Sentinel HASP Master key is used by your production staff to create licenses and lock them to Sentinel HASP protection keys, to write specific data to the memory of a Sentinel HASP protection key, and to update licenses already deployed in the field. The Sentinel HASP Master key is used in conjunction with Sentinel HASP Business Studio.

End-User Keys

Two types of Sentinel HASP protection keys are available—HASP HL keys, which are physical USB keys that connect to a computer, and HASP SL keys, which are virtual keys that lock your software to a specific machine. Your software and the user license is locked to the Sentinel HASP protection key that you select.

All HASP HL keys—with the exception of HASP 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

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

HASP HL Keys

HASP HL keys are distributed with your software to end users. The keys connect to the end users' computers. A variety of HASP HL keys are available to suit your requirements.

HASP 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 HASP uses *LicenseOnChip* technology to protect HASP HL keys against license tampering.

HASP 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.

For information about the available HASP HL keys, see [Available Sentinel HASP Protection Keys on page 42](#).

HASP SL Keys

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

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 the Sentinel HASP Business Studio Server, together with the fingerprint of the machine. Sentinel HASP Business Studio validates 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 HASP SL key, which is installed on the end user's machine. This process is also used for updating license terms.

For information about the attributes of HASP SL keys, see [Available Sentinel HASP Protection Keys on page 42](#).

For information about how Sentinel HASP prevents tampering of time-based licenses locked to HASP SL keys, see *How Sentinel HASP Protects Time-based Licenses Locked to HASP SL Keys* [on page 343](#).

Available Sentinel HASP Protection Keys

The following table lists the available Sentinel HASP protection keys and their attributes.

Note:

For full technical specifications of the keys, refer to the relevant product data sheet.

Sentinel HASP Protection Key	Description	Supported Automatic Licensing and License Terms	Memory Available for Software Vendor
HASP HL Basic	Hardware-based AES encryption for copy and IP protection.	<ul style="list-style-type: none"> ■ Perpetual 	—
HASP HL Pro	Hardware-based AES encryption for copy and IP protection. Supports automatic licensing for multiple applications/Features.	<ul style="list-style-type: none"> ■ Perpetual ■ Feature-based ■ Per-use ■ Demo ■ License terms determined by counters 	<ul style="list-style-type: none"> ■ Read/write 112 bytes ■ ROM 112 bytes
HASP HL Max	Hardware-based AES encryption for copy and IP protection. Supports automatic licensing for multiple applications/Features.	<ul style="list-style-type: none"> ■ Perpetual ■ Feature-based ■ Per-use ■ Demo ■ License terms determined by counters 	<ul style="list-style-type: none"> ■ Read/write 4 KB ■ ROM 2 KB
HASP HL Drive	Hardware-based AES encryption for copy and IP protection. Supports automatic licensing for multiple applications/Features.	<ul style="list-style-type: none"> ■ Perpetual ■ Feature-based ■ Per-use ■ Demo ■ License terms determined by counters 	<ul style="list-style-type: none"> ■ Flash memory 512 MB/2 GB ■ Read/write 4 KB ■ ROM 2 KB
HASP HL Time	Hardware-based AES encryption for copy and IP protection. Supports automatic licensing for multiple applications/Features. Contains internal real-time clock.	<ul style="list-style-type: none"> ■ Perpetual ■ Feature-based ■ Rental ■ Subscription ■ Demo ■ License terms determined by time and date of internal real-time clock 	<ul style="list-style-type: none"> ■ Read/write 4 KB ■ ROM 2 KB

Sentinel HASP Protection Key	Description	Supported Automatic Licensing and License Terms	Memory Available for Software Vendor
HASP HL Net	Hardware-based AES encryption for copy and IP protection. Supports automatic licensing for multiple applications/Features.	<ul style="list-style-type: none"> ■ Perpetual ■ Feature-based ■ Floating ■ Per-use ■ Demo ■ License terms determined by number of users and counters 	<ul style="list-style-type: none"> ■ Read/write 4 KB ■ ROM 2 KB
HASP HL NetTime	Hardware-based AES encryption for copy and IP protection. Supports automatic licensing for multiple applications/Features. Contains internal real-time clock.	<ul style="list-style-type: none"> ■ Perpetual ■ Feature-based ■ Rental ■ Subscription ■ Floating ■ Per-use ■ Demo ■ License terms determined by number of users and time and date of internal real-time clock 	<ul style="list-style-type: none"> ■ Read/write 4 KB ■ ROM 2 KB
HASP SL	Software-based AES encryption for copy and IP protection. Supports automatic licensing for multiple applications/Features.	<ul style="list-style-type: none"> ■ Perpetual ■ Feature-based ■ Rental ■ Subscription ■ Floating ■ Per-use ■ Demo ■ Volume ■ License terms determined by number of users, counters, and time and date of system clock 	<ul style="list-style-type: none"> ■ Read/write 4 KB ■ ROM 2 KB

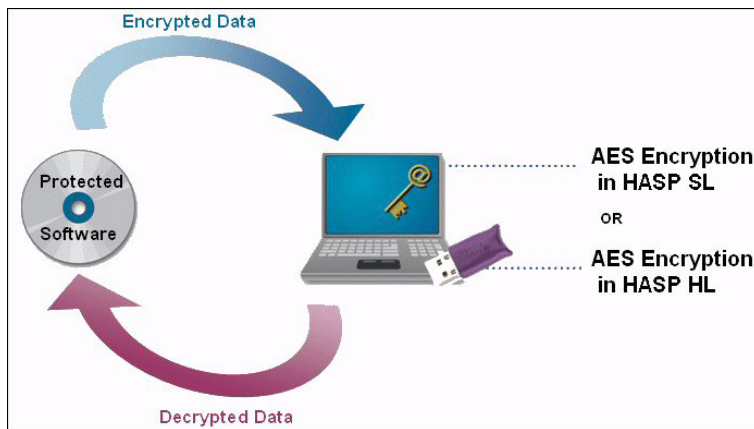
Sentinel HASP Protection Process

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

Encryption and Decryption

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

The following diagram illustrates how encrypted data in your protected software is decrypted in the Sentinel HASP protection key and returned unencrypted to the software.



Obtaining Additional Information About Sentinel HASP

This chapter has provided an overview of the major concepts and principles of Sentinel HASP, and the comprehensive protection and licensing solution that Sentinel HASP provides.

The remainder of this guide explains in detail how you can best use the many elements of Sentinel HASP to meet your company's software protection, licensing, and distribution requirements.

Additional information is available in the Help documentation for the various Sentinel HASP tools, and in additional Sentinel HASP documentation that you can download using the following URL:

<http://www3.safenet-inc.com/support/hasp-srm/vendor.aspx>

Part 2 Protection

In this section:

- **Chapter 2: Protecting Software**

Provides an overview of Sentinel HASP software protection, including its fundamental elements, a summary of how it works, and an introduction to Sentinel HASP protection methods.

- **Chapter 3: Sentinel HASP Run-time API Protection**

Provides an overview of the Sentinel HASP Run-time API, details the prerequisites for using the API, introduces the Sentinel HASP ToolBox application and describes the functionality of the API.

- **Chapter 4: Sentinel HASP Envelope Protection**

Provides an overview of software protection using Sentinel HASP Envelope, details the prerequisites for using the application, and describes its functionality. In addition, it describes the Sentinel HASP Envelope protection parameters and how to encrypt data files.

- **Chapter 5: Protection Strategies**

Outlines strategies for maximizing Sentinel HASP protection, including best practices and optimizing the use of the Sentinel HASP Run-time API and Sentinel HASP Envelope.

- **Chapter 6: Working with the DataHASP Encryption Utility**

Describes data file protection using the DataHASP utility. It includes information about who should encrypt data files and when they should be encrypted.



Protecting Software

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

In this chapter:

- Sentinel HASP Protection

Sentinel HASP Protection

Sentinel HASP 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 Sentinel HASP-protected program requires access to a specific Sentinel HASP protection key in order to run. The protected program queries the Sentinel HASP protection key for pre-defined information. If the Sentinel HASP 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 HASP protection method, implementation is straightforward. Regardless of the selected protection strategy, protected programs only work correctly if they can access the information stored in a specific Sentinel HASP protection key.

Elements of Sentinel HASP Protection

The Sentinel HASP protection system is based on the following:

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

Protecting Programs and Data Files

Sentinel HASP provides two primary protection methods:

- Sentinel HASP Envelope
- Sentinel HASP Run-time 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 HASP protection key.

What can be Protected

Sentinel HASP 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 HASP 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 chapters:

- Chapter 3, *Sentinel HASP Run-time API Protection*
- Chapter 4, *Sentinel HASP Envelope Protection*
- Chapter 6, *Working with the DataHASP Encryption Utility*

Identifying the Sentinel HASP Software Protection Key

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

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

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

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

AES Encryption

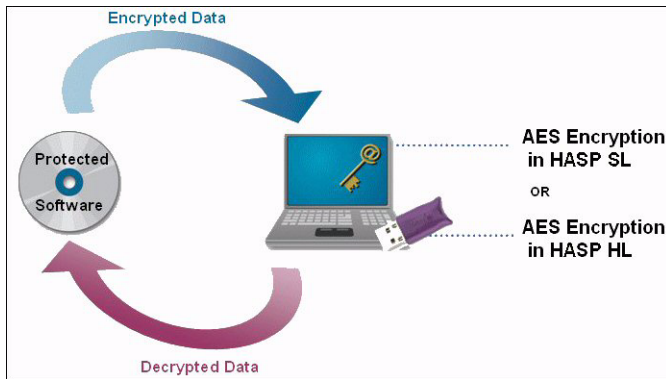
A protected program relies on the ‘intelligence’ in the memory of a specific Sentinel HASP protection key in order to function. In addition to the checks for the Sentinel HASP protection key, data can be encrypted and decrypted using the intelligence available in the Sentinel HASP protection key.

AES Encryption and Decryption

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

Your protection schemes should always involve greater sophistication than merely confirming the presence of the required Sentinel HASP protection key. However, verifying the required Sentinel HASP 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 HASP protection key, where it is decrypted.

If the data is correct, the Sentinel HASP protection key is considered to be “present.” This protection is graphically illustrated the following figure. For additional information, see [Time Functions on page 69](#).



Encryption/Decryption of Data

Confidential Protection Parameters

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

Vendor Code

Each Sentinel HASP customer is assigned a unique *Vendor Code* that must be kept confidential. The Vendor Code forms an integral part of the protection parameters that constitute the inherent link between the protected programs and the Sentinel HASP protection key. However, the Vendor Code is only part of the link. The code on its own is insufficient to prevent illegal use of the software. It merely provides the protected software with access to the Sentinel HASP protection key and its resources.

All Sentinel HASP protection applications require the Vendor Code. For information on how to access the code, see [Extracting the Vendor Code from Sentinel HASP Vendor Keys on page 60](#).

Utilizing HASP Memory

The HASP memory on Sentinel HASP 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 HASP memory, including snippets of program code, customer name, or any other data.

Use the memory editors included in Sentinel HASP ToolBox to read or write data in the HASP memory. For additional information, see [Memory Functions on page 69](#).

Anti-Debugging and Reverse Engineering Measures

Sentinel HASP 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 HASP Envelope implements contingency measures to ward off such attacks and prevent hackers from uncovering algorithms used inside protected software.

Selecting a Protection Method

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

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

- What the Sentinel HASP protection key should protect
- How the Sentinel HASP 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 HASP, there are various options for applying protection. Sentinel HASP Run-time API is used to protect the software before it is compiled. Protection can also be applied after the software is compiled using Sentinel HASP 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 HASP Run-time API or the Sentinel HASP Envelope protection method, or both, depending on your specific requirements. Use the following table to determine which method best meets your specific requirements.

Sentinel HASP Envelope	Sentinel HASP Run-time API
<ul style="list-style-type: none">■ Quick, automatic protection process that shields your software■ Define specific protection parameters for your programs■ No source code required■ Anti-debugging and reverse engineering measures provided	<ul style="list-style-type: none">■ Manual implementation of calls to Sentinel HASP Run-time API■ Controlled process ensuring maximum security. The strength of protection is proportional to the degree to which the Sentinel HASP Run-time API's functionality is invested in implementation.■ Source code must be available■ Maximum flexibility

Importance of Control over the Protection Scheme

When applying protection using Sentinel HASP Run-time API, you control the entire protection process. You determine when the protected program queries the Sentinel HASP protection key, and how it should behave in different scenarios. With Sentinel HASP Envelope, compiled programs are wrapped with random protection parameters. If you run Sentinel HASP 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 HASP 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 HASP Run-time 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 HASP Run-time API.

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

Note:

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

Sentinel HASP Run-time API Protection

This chapter describes the Sentinel HASP Run-time API protection method.

In this chapter:

- Overview
- Sentinel HASP Run-time API Prerequisites
- Learning the Sentinel HASP Run-time API
- Implementation
- Sentinel HASP Run-time API Functionality

Overview

The Sentinel HASP Run-time application programming interface (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 HASP Run-time API is utilized, determines the overall level of software security. To fully utilize the protection offered by the Sentinel HASP Run-time 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 HASP Run-time API. For a description of the functions that make up the Sentinel HASP Run-time API, see Appendix B, *Sentinel HASP Run-time API Reference*.

To protect your software using the Sentinel HASP Run-time API, you insert calls to a Sentinel HASP protection key throughout your application's source code. You can add calls to your application that check for the presence of a Sentinel HASP protection key at any point during run-time, and you can designate responses to these checks. For example, if the required Sentinel HASP 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 HASP protection key for specific data. In addition, you can use the Sentinel HASP Run-time API to encrypt or decrypt data.

To facilitate a speedy learning curve, we recommend that you familiarize yourself with and test specific Sentinel HASP Run-time API functions using Sentinel HASP ToolBox. Sentinel HASP ToolBox is a GUI-based application that interfaces with the Sentinel HASP Run-time API. For additional information, see [Learning the Sentinel HASP Run-time API](#) on page 62.

Sentinel HASP also includes Sentinel HASP Run-time API sample folders for specific compilers. Each Sentinel HASP interface includes a sample application demonstrating API usage and a specific header file. The sample applications are located in the **Samples** folder in the Windows directories on the Sentinel HASP installation DVD.

Universal Sentinel HASP Run-time API

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

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

Sentinel HASP Run-time API Prerequisites

You must install the Sentinel HASP Run-time Environment to enable the Run-time API. For additional information, see the *Sentinel HASP Installation Guide*.

Vendor Code

It is necessary to provide the Vendor Code in order to access a Sentinel HASP protection key and its resources, including memory. Vendor Codes are usually stored in the **VendorCodes** folder. On most Windows installations, the directory is located at:

```
...\Documents and Settings\[logged_in_user_name]\My Documents\  
Aladdin\Sentinel HASP [version]\VendorCodes
```

In the Sentinel HASP Developer Kit, customers are provided with HASP HL Demo keys that work with the DEMOMA Vendor Code. This Vendor Code can be used to apply protection with the Sentinel HASP Run-time API.

Note:

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

The first time you order Sentinel HASP protection keys, you also receive two Sentinel HASP Vendor keys—a Sentinel HASP Developer key and a Sentinel HASP Master key—that contain your company's unique confidential Vendor Code. The Sentinel HASP Developer key is used by engineers for adding protection to your software. The Sentinel HASP Master key is used for producing licenses and orders.

Sentinel HASP Vendor Suite applications (Sentinel HASP Envelope, Sentinel HASP ToolBox, and Sentinel HASP Business Studio) must recognize and have access to the unique Vendor Code that was assigned to you when your first order was supplied by SafeNet. The Vendor Code is stored inside your Sentinel HASP Vendor keys. Sentinel HASP Vendor keys are introduced using the MasterHASP Wizard, as described in the following section.

Note:

If you have already introduced your Sentinel HASP Developer key, it is usually not necessary to re-introduce it.

Extracting the Vendor Code from Sentinel HASP Vendor Keys

You need to extract the Vendor Code from your Sentinel HASP Vendor keys so that the Sentinel HASP system will recognize it when you are working with any of the Vendor Suite applications.

Depending on your Sentinel HASP configuration, if you launch a Sentinel HASP Vendor Suite application, and you have connected a new Sentinel HASP Vendor key to your computer, the MasterHASP Wizard will launch automatically. Alternatively, use the following procedure to extract the information.

To extract the Vendor Code:

1. Connect your Sentinel HASP Master key to your computer.
2. Run the MasterHASP Wizard (**Start > Programs > SafeNet > Sentinel HASP > Tools > MasterHASP Wizard**). The MasterHASP Wizard launches, detects, and lists all new Sentinel HASP Vendor keys (Master key and Developer key).

Note:

If you open either Sentinel HASP Envelope or Sentinel HASP ToolBox, and the application detects a new Sentinel HASP Vendor key, the MasterHASP Wizard will launch automatically.

3. Enter a name for the file in which the Vendor Code information will be saved. It is recommended that you store all the Vendor Codes in the VendorCodes folder. On most Windows installations, the directory is located in:

...\\Documents and Settings\\[logged_in_user_name]\\My Documents\\
Aladdin\\Sentinel HASP [version]\\VendorCodes

By default, the Sentinel HASP Vendor Suite applications search this directory for the Vendor Code.

4. In the Specify API Settings window of the wizard, select the libraries for which you want to generate APIs. If you want to merge APIs of multiple Batch Codes into a single library, click **Advanced**. You can merge up to four APIs to a single library. When you merge APIs, individual libraries are generated in addition to the merged ones.

The generated APIs are located in the following directories, as appropriate:

...\Documents and Settings\[logged_in_user_name]\My Documents\
Aladdin\Sentinel HASP [version]\API\Runtime\C

...\Documents and Settings\[logged_in_user_name]\My Documents\
Aladdin\Sentinel HASP [version]\API\Runtime\COM

...\Documents and Settings\[logged_in_user_name]\My Documents\
Aladdin\Sentinel HASP [version]\API\Runtime\DotNet

...\Documents and Settings\[logged_in_user_name]\My Documents\
Aladdin\Sentinel HASP [version]\API\Runtime\Java

...\Documents and Settings\[logged_in_user_name]\My Documents\
Aladdin\Sentinel HASP [version]\API\Runtime\Delphi

5. When prompted, update the vendor library. This library is required for HASP SL protection, including creating trialware.
6. By default, your Vendor Code information is saved in the following directory:

...\Documents and Settings\[logged_in_user_name]\My Documents\
Aladdin\Sentinel HASP [version]\API\Runtime\VendorCodes

Vendor-specific File Naming Conventions

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

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

The format of API library names (for Windows) is `hasp_windows_[language]_[vendorcode].[library extension]`. For example, **hasp_windows_demo.lib** is a C-language API library associated with a demo key.

Learning the Sentinel HASP Run-time API

There are two components of Sentinel HASP that enable you to study how the Sentinel HASP Run-time API works, and its range of capabilities.

- **Sentinel HASP ToolBox:** A utility with a graphic user interface that is part of Sentinel HASP Vendor Suite. For more information, see [Sentinel HASP ToolBox on page 62](#).
- **Sentinel HASP Run-time API Samples:** A set of examples for implementing the Sentinel HASP Run-time API. For more information, see [Sentinel HASP Run-time API Samples on page 63](#).

Sentinel HASP ToolBox

Sentinel HASP ToolBox is an interactive interface to the Sentinel HASP Run-time API. You execute calls to the Sentinel HASP Run-time API via Sentinel HASP ToolBox that are then relayed to a Sentinel HASP protection key.

To use Sentinel HASP ToolBox you must have a Sentinel HASP Developer key and a valid Vendor Code so that you can access Sentinel HASP protection keys. The program is activated from Sentinel HASP Vendor Suite. For specific information on how to use Sentinel HASP ToolBox, see the application's Help documentation.

API-related Functionality

Sentinel HASP ToolBox acts like a tutorial for the Sentinel HASP Run-time API. Its 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 the Sentinel HASP Run-time API.

Every Sentinel HASP Run-time API function included in Sentinel HASP 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 HASP protection key.
The program can also be used to decrypt data buffers.
- Create multiple programming language interfaces for the Sentinel HASP Run-time API.

Sentinel HASP Run-time API Samples

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

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

Note:

See the Sentinel website and the Sentinel HASP Installation DVD for information on available samples for specific programming languages.

Implementation

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

Planning Your Requirements

Before implementing the Sentinel HASP Run-time API, the following preliminary issues should be considered.

- **What do you want to protect?**

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

Note:

Feature protection is handled through the Feature ID parameter. For additional information, see [hasp_login_scope\(\) on page 319](#).

- **Will encrypted data be included in my implementation scheme?**

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

- **Is data going to be stored in the Sentinel HASP protection key memory?**

If the software is protected by a Sentinel HASP protection key with memory functionality, sensitive data can be stored in the Sentinel HASP protection key. The Sentinel HASP Run-time API enables access to Read or Write to Sentinel HASP protection key memory. Use Sentinel HASP ToolBox to write data buffers to Sentinel HASP protection key memory.

Sentinel HASP Run-time 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 HASP Run-time API.

The recommended workflow for implementing the Sentinel HASP Run-time 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 HASP 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 [hasp_login_scope\(\)](#) on page 319.

3. After a login session is established, you can use other Sentinel HASP Run-time API functions to communicate with the Sentinel HASP protection key. For example, you can use the `hasp_decrypt` function to decrypt important data used by your application. You can also read data stored in the Sentinel HASP protection key memory, set timestamps, and so on.
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.

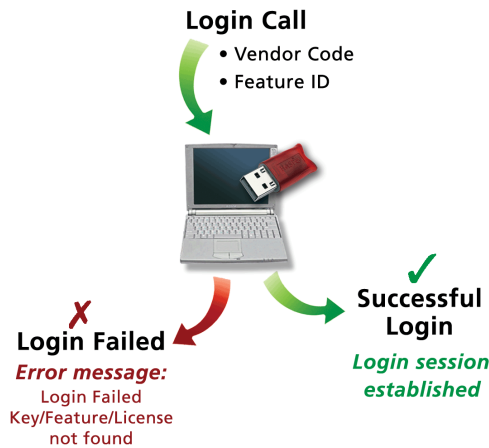
Note:

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

Sentinel HASP Run-time API Login Function

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

If the Sentinel HASP 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 HASP protection key.



Sentinel HASP Login Operation Summary

Login Options

When using Sentinel HASP Run-time API implementation, login calls are not dependent on specific Sentinel HASP 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 HASP protection key
- How the login counter should be handled
- Whether to enable or disable connection to the Sentinel HASP 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 HASP 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 HASP 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 HASP Run-time API, as follows:

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

Search Options

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

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

Login Counter

By default, when a Sentinel HASP license is accessed in a HASP HL Net key, license usage is determined per workstation. You can override this condition so that license usage is counted per process. This implies that the license counter is decremented per process.

Access to Legacy Memory on HASP HL Key

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

Note:

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

Sentinel HASP Run-time API Functionality

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

This section describes the Sentinel HASP Run-time API options that are available after a successful login session is established. For a detailed discussion about how to optimize your Sentinel HASP Run-time API implementation, see Chapter 5, [Protection Strategies on page 101](#). For a demonstration of how the Sentinel HASP Run-time API works, use Sentinel HASP ToolBox. All functionality described in this section is detailed in Appendix B, *Sentinel HASP Run-time API Reference*.

Function Groups

Sentinel HASP Run-time 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 HASP protection key. For more information about logging in, see [Login Options on page 66](#). 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 HASP 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 HASP Run-time API can be used to:

- Read data buffers stored in the Sentinel HASP protection key memory
- Write data buffers to the Sentinel HASP protection key memory

The size of the data buffers is restricted by the memory available in the specific Sentinel HASP protection key type. [Available Sentinel HASP Protection Keys on page 42](#) lists the memory capacity available for each Sentinel HASP protection key type.

Time Functions

If you are using a HASP HL Time key or HASP HL NetTime key, the Sentinel HASP Run-time API can be used to access the real-time clock in the key. This functionality enables you to read the time. Two date and time conversion functions are included in the Sentinel HASP Run-time API.

Management Functions

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

The detach function enables you to detach a license from a pool of network seats when using HASP SL keys.

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 [*hasp_update\(\)* on page 324](#). This function is the main facilitator of Sentinel HASP Remote Update System.

Sentinel HASP Envelope Protection

This chapter describes software protection using Sentinel HASP Envelope.

In this chapter:

- [Functionality](#)
- [Sentinel HASP Envelope for Windows](#)
- [Accessing and Encrypting Data Files for Windows Programs](#)
- [Protecting .NET Assemblies](#)
- [Sentinel HASP Envelope for Linux Applications](#)
- [Sentinel HASP Envelope for Mac Binaries](#)
- [Sentinel HASP Envelope for Java Executables](#)

Functionality

Sentinel HASP 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 HASP 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 HASP Envelope can also be used to protect Mac executables and dynamic shared libraries (Mach-O), and Linux executables and shared objects. For more information, see [Sentinel HASP Envelope for Mac Binaries on page 96](#), and [Sentinel HASP Envelope for Linux Applications on page 93](#).

Note:

The word *program* is used throughout this chapter as a generic reference to the various file types that can be protected using Sentinel HASP Envelope, regardless of whether they are executables, binaries, assemblies, libraries or shared objects.

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

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

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

- Protect Win32, Windows x64, 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 32-bit and 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 HASP protection key
- Customize the run-time messages that will be displayed to end users running protected programs

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

Note:

The random multi-layer wrapping of protected applications by Sentinel HASP Envelope ensures that implemented protection strategies differ from one protected program to another.

Command line utilities enable you to protect:

- Win32, Windows x64, and .NET executables and DLL files
- Java executables
- 32-bit and 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 HASP Envelope GUI. This simplifies the process of reapplying protection parameters to your program during the development process.

Basic Protection Workflow

This section provides a workflow that describes the elements of protecting applications using Sentinel HASP Envelope. Additional information about specific procedures is provided in the Sentinel HASP Envelope Help documentation.

1. Launch the Sentinel HASP Envelope graphical interface from Sentinel HASP Vendor Suite.
2. Add the executable, library, or .NET assembly you want to protect to the project.
3. Define protection parameters for the protected program.
4. Protect the program.

5. Distribute the protected software together with your encrypted Sentinel HASP protection keys.
-

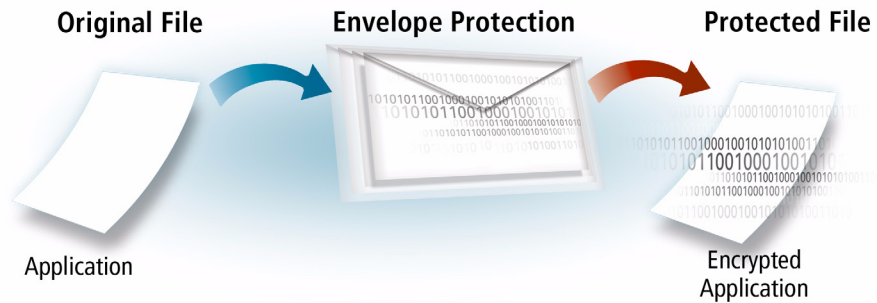
Note:

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

Sentinel HASP 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 HASP protection key. If the Sentinel HASP protection key is present, the protected file runs.

The logic of Sentinel HASP 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.



Note:

- To ensure the highest level of security for your software, Sentinel HASP 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 HASP Envelope.
-

Required and Optional Protection Parameters

This section outlines the mandatory and customizable parameters that can be specified for protecting software with Sentinel HASP Envelope.

Mandatory Parameters

The following information must be provided in order to protect software using Sentinel HASP 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 `[user's home directory]\Aladdin\Sentinel HASP [version]\VendorTools\VendorSuite\Protected`
- **Vendor Code:** You must provide a valid Vendor Code in order to access a Sentinel HASP protection key. On initial activation of Sentinel HASP Envelope, the default Vendor Code is specified as **DEMOMA**. Select your Vendor Code in the Sentinel HASP Profile screen.

This information is sufficient to protect a program.

General Customizable 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 on page 77](#).
- **HASP search mode:** You can determine where a protected program searches for the Sentinel HASP protection key. For additional information, see [Searching for a Sentinel HASP Protection Key on page 77](#).

WARNING!

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

Searching for a Sentinel HASP Protection Key

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

The following options are available:

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

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 HASP Envelope is Feature ID 0.

For additional information on Features and licensing, see [Identifying Functional Components \(Features\) on page 132](#) and [Managing Features on page 143](#).

When you protect a Win32, Windows x64, Mac or Linux application with Sentinel HASP 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 HASP Run-time API. For additional information, see Chapter 3, [Sentinel HASP Run-time API Protection](#).

Protecting .NET Assemblies

When you protect a .NET assembly with Sentinel HASP 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 86](#).
- Method-specific Features. For additional information, see [Method-specific Features and Parameters in .NET Assemblies on page 88](#).

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

Sentinel HASP Envelope for Windows

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

Prerequisites for Windows

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

- Sentinel HASP Run-time Environment
- Sentinel HASP Vendor Suite
- A valid Vendor Code stored in the **VendorCodes** folder.
For additional information, see *Extracting the Vendor Code from Sentinel HASP Vendor Keys* on page 60.
- `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 HASP Envelope

In the **Start** menu, select **Programs > SafeNet > Sentinel HASP > Vendor Suite**. From the Sentinel HASP Vendor Suite program selection screen, launch Sentinel HASP Envelope.

Sentinel HASP Envelope Protection Parameters

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

Sentinel HASP 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 HASP Envelope Help documentation.

This section provides an overview of the Sentinel HASP 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 Parameters on page 76](#). Other common parameters are described in [General Customizable Parameters on page 76](#).

Sentinel HASP 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 Win32 or Windows x64 Programs on page 80](#), and [Accessing and Encrypting Data Files for Windows Programs on page 81](#).
- For additional information about settings for .NET assemblies, see [Protecting .NET Assemblies on page 85](#), and [Code and Symbol Obfuscation in .NET Assemblies on page 90](#).
- For additional information about settings for Java executables, see [Protecting Java Executables on page 100](#).

Protecting Win32 or Windows x64 Programs

When you protect a Win32 or Windows x64 program with Sentinel HASP Envelope, you can determine protection attributes and aspects of the behavior of the protected program.

Protected Program Behavior

Sentinel HASP 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 HASP protection key. These queries include random encryption and decryption procedures.
- The time interval between checks for the presence of a required Sentinel HASP 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 HASP 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 HASP system to block potential attacks intended to undermine the protection scheme.
- Defining the number of protective module layers that are wrapped around a protected application. Possible values range from 1 to 50. The default setting is 12.

Note:

Increasing the number of protective modules increases the startup time for a protected application, and the resultant file size. There is also a trade-off between encryption level and protected file size and startup speed. A higher encryption level causes a slower startup, and increases the size of the protected application.

- Specifying the frequency of Sentinel HASP protection key access for encryption. The parameter controls the compactness of the Sentinel HASP protection key calls made by the protected application. An **Encryption Level** slider is provided to specify the frequency of Sentinel HASP protection key access for encryption.

Accessing and Encrypting Data Files for Windows Programs

If your protected program (Win32 or Windows x64) accesses separate data files, you can use Sentinel HASP to protect these files, in addition to envelope protection of the program itself, as follows:

- You can control if and when the protected program accesses such data files. For additional information, see [Data File Handling on page 81](#).
- You can apply Sentinel HASP encryption to the data files. For additional information, see [Encrypting Data Files on page 82](#).

Data File Handling

Your protected Windows program may require access to data files during run-time. Sentinel HASP Envelope enables you to control the access of data files by the protected software. Sentinel HASP Envelope offers the following control mechanisms:

- **Data filters:** You can determine what file types can be accessed at run-time by the protected software. You can also determine what files to exclude.
- **Data encryption keys:** Eight printable characters used to create an encryption key for encrypting and decrypting data files.

Note:

Use the same encryption key when multiple applications access the same document set.

Sentinel HASP Envelope includes the data filters and encryption key information as part of the protection scheme applied to the protected application. The encryption of data files is handled by the DataHASP utility. For additional information, see [Working with the DataHASP Encryption Utility on page 109](#). Alternatively, you can use a command-line utility. For additional information, see [Using dfcrypt.exe on page 82](#).

Setting Data File Filters

Sentinel HASP Envelope enables you to create file filters for protected Windows programs. File filters determine which data files and data file types can work in conjunction with protected programs. For additional information, see the Help documentation.

Run-time User Support

You can customize run-time messages for end users who are using applications protected by Sentinel HASP Envelope. Sentinel HASP 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.

Encrypting Data Files

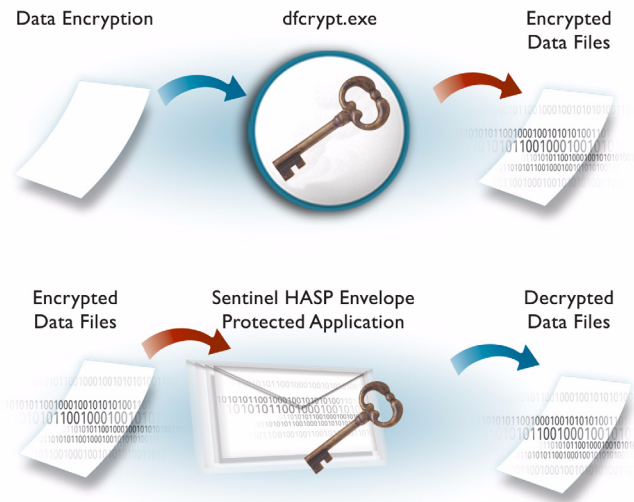
Windows programs protected with Sentinel HASP Envelope can access encrypted data files that are defined by data filters and use a specific encryption key. Sentinel HASP Envelope itself does not encrypt data files. However, you can encrypt data files using the DataHASP interface, or you can use the `dfcrypt.exe` command-line utility. For additional information about using the DataHASP interface, see Chapter 6, *Working with the DataHASP Encryption Utility*.

Using `dfcrypt.exe`

Instead of using the DataHASP utility, you can use a command-line to encrypt data files. The `dfcrypt.exe` utility generates data files that can be processed by executables protected by Sentinel HASP Envelope.

The command-line utility requires a specific set of input parameters to function. After these parameters have been applied to a defined set of data files, the encrypted files can be accessed by Sentinel HASP Envelope.

The following figure illustrates the workflow for data file handling using `dfcrypt.exe`. After data files are encrypted, applications protected with Sentinel HASP Envelope can decrypt and access the files. This is only possible if, at protection time, you specify the encryption key in Sentinel HASP Envelope.



Data File Handling with *dfcrypt.exe*

To use *dfcrypt.exe*, specify the following:

- A command switch
- A list of source files and directories
- A destination for the encrypted output

When specifying multiple source files or a directory, the destination input specified must be an existing directory.

The following format must be used to input parameters:

```
dfcrypt<option> source destination
```

For example:

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

Available dfcrypt.exe Commands

The following table lists the available `dfcrypt.exe` commands.

Command	Action
<code>-e, --encrypt</code>	Encrypts data, available by default
<code>-d, --decrypt</code>	Decrypts data
<code>-c, --vcf:<file></code>	Specifies a Vendor Code file (mandatory)
<code>-k, --key:<key></code>	Specifies an encryption key to be used to encrypt data files. Must contain 8 printable characters. (mandatory)
<code>-o, --overwrite</code>	Overwrites destination files
<code>-r, --recursive</code>	Enables recursive handling of subdirectories
<code>-h, --help</code>	Displays the help screen, listing <code>dfcrypt.exe</code> commands
<code>-q, --quiet</code>	Suppresses output by excluding copyright information and the progress indicator. Only error messages are displayed. This is particularly useful in Makefile integration.

Running Sentinel HASP Envelope from a Windows Command-line

Sentinel HASP 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 HASP Envelope is primarily used for automated processes. Before running the command-line utility, create and save protection projects together with their configuration files using `envelope.exe`.

To access the command-line version of Sentinel HASP Envelope, go to ...\\Program Files\\SafeNet\\Sentinel HASP\\VendorTools\\VendorSuite\\envelope.com

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

Command-line Options

The following parameters are available for use with the Sentinel HASP Envelope command-line version:

Command	Description
-h /--help	Displays the list of command-line parameters. Press Enter to return to the command-line console.
-p /--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.
<project>	The command-line version starts the GUI version with the specified project running as the current project.
-a / -- accesscode <code><project>	Specifies the access code for the connected HASP Vendor key. This switch is only required if an access code has been specified using the MasterHASP wizard.

Protecting .NET Assemblies

Sentinel HASP 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.

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 on page 86](#).

When you protect a .NET assembly with Sentinel HASP Envelope, you specify a global Feature that protects the entire assembly. For additional information, see [Global Features in .NET Assemblies on page 86](#).

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 88.

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 90.

.NET Considerations

When protecting .NET assemblies, consider the following issues:

- You must protect your assemblies in a development environment. Sentinel HASP Envelope requires libraries that are not part of the .NET framework, but are included in the development environment.
- Sentinel HASP Envelope for .NET requires access to all assemblies and their dependencies.
- Sentinel HASP Envelope breaks the strong name signature of signed assemblies. You can choose to re-sign the assembly in Sentinel HASP Envelope, as part of the protection process.
- When you protect a .NET Framework 1.x assembly, the Sentinel HASP 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 HASP DLL is required. For more information, see “*Protection-related Software*” on page 208

Global Features in .NET Assemblies

When you protect a .NET assembly with Sentinel HASP 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 HASP 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 HASP 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 Win32 shell** or **Only Windows x64 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 HASP 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 program logs into an individual method

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

You can determine how often the protected program logs into each Feature ID in the Sentinel HASP 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
Once per program (Default)	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.
Once per class instance	<p>A check is performed when the method is run, once for each Feature ID within the same class.</p> <ul style="list-style-type: none"> ■ If the same Feature ID is also assigned to the class constructor, the check is performed the first time the .ctor method is run. ■ If the same Feature ID is used in other classes, the check is performed separately for each class. <p>Note: The Once per class instance frequency is available only for Instance methods.</p>
Every time	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 a counter-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 HASP protection key.

If methods that do not have a valid license in a HASP protection key are called, it will cause Sentinel HASP 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 HASP installation that enables you to ensure that the error loop does not occur. The API is located in

...\Program Files\SafeNet\Sentinel HASP\Samples\Envelope\EnvelopeRuntime.NET.

To use the API

To prevent an error loop occurring if a method for which a license has expired or does not exist, register a **NotificationDelegate** handler, as follows:

1. Include the **Aladdin.HASP.EnvelopeRuntime** assembly in your source code.
2. Select one of the following handlers, depending on the behavior that you require when the handler is invoked.

EnvelopeRuntimeStatus.StatusThrowException	Discontinue execution path by throwing an exception
EnvelopeRuntimeStatus.StatusReturnNothing	Discontinue execution path by returning a null reference (Nothing)
EnvelopeRuntimeStatus.StatusAlertAndRetry	Default Display message box that asks the user for the license, then retry.
EnvelopeRuntimeStatus.StatusRetry	Transparently retry

The **NotificationDelegate** handler will be invoked whenever the Sentinel HASP Envelope run-time cannot decrypt a protected method. It will also receive the appropriate Sentinel HASP error status code. You can use this information in your own “license required” error message, instructing your user to abort or retry their action.

Code and Symbol Obfuscation in .NET Assemblies

Obfuscation is the process of turning meaningful strings into random strings of letters or numbers. Using Sentinel HASP, 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.

Defining Sentinel HASP Envelope Protection Settings in Source Code

Instead of specifying your protection settings using the Sentinel HASP 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.

Protection settings that are defined in your source code override settings already specified in the Sentinel HASP Envelope GUI, and you cannot use the GUI to edit the settings once they have been integrated in your code. However, you can use the GUI to view the protection settings that have been specified in your code.

To use the custom attributes, you must include the **`Aladdin.HASP.Envelope`** assembly in your project.

Note:

`Aladdin.HASP.Envelope.dll` is not required for protected assemblies and does not have to be distributed to your customers.

Using the custom attributes, you can define the following protection settings:

Name	Description	Type	Default Value
Protect	Protect this method	bool	true
FeatureId	Feature ID to be used for protection	int	-1 (global Feature)
Encrypt	Encrypt this method	bool	true
Code Obfuscation	Obfuscate complete code for this method	bool	false
Frequency	When license for method is to be checked	EnvelopeMethod ProtectionFrequency	CheckOncePerApplication

Sample

The following sample shows how protection settings can be applied in source code. The source code comments are italicized.

```
using System;
using Aladdin.HASP.Envelope;

/// do not protect any methods in this assembly
[assembly: EnvelopeMethodProtectionAttributes(Protect=false)]
namespace MyProgram
{
    /// methods in this class do not get protected, because
        protection
    /// settings are inherited from assembly
    class MyExample
    {
        /// this method does not get protected
        static void Main(string[] args)
        {
            Console.WriteLine("{0} + {1} = {2}", 3, 4,
                MyClass.Add(3, 4));
            Console.WriteLine("{0} * {1} = {2}", 3, 4,
                MyClass.Multiply(3, 4));
        }
    }

    /// protect all methods in this class with Feature ID 0
    [EnvelopeMethodProtectionAttributes(Protect=true,
        FeatureId=0)]
    class MyClass
    {
        /// protect the Add method using Feature ID 1,
            obfuscate the code,
        /// check the license everytime this method is
            invoked
        [EnvelopeMethodProtectionAttributes(Protect=true,
            FeatureId=1, Encrypt=true,
            CodeObfuscation=true,
            Frequency=EnvelopeMethodProtectionFrequency.
                CheckEveryTime)]
        public int Add(int a, int b)
        {
            return a + b;
        }

        /// protection settings for this method are
            inherited from the settings of the class
        public int Multiply(int a, int b)
        {
            return a * b;
        }
    }
}
```

Additional Sentinel HASP Envelope .NET API Information

- The protection definitions and defaults are based on those provided in the Sentinel HASP Envelope GUI
- By default, static methods and methods that have a very small footprint are not protected. To protect these methods, you must specify the protection settings at method level in your source code.

Sentinel HASP Envelope for Linux Applications

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

Sentinel HASP Envelope Prerequisites for Linux

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

- Sentinel HASP Run-time Environment
- Sentinel HASP Vendor Suite, containing the Sentinel HASP Envelope command-line utility for Linux and the MasterHASP wizard
- A valid Vendor Code stored in the **VendorCodes** folder. For additional information, see *Extracting the Vendor Code from Sentinel HASP Vendor Keys* on page 60.
- The Linux executables and shared libraries that you want to protect. Both x86 and x86_64 ELF executables and shared libraries are supported.

Protecting Linux Applications

You protect Linux applications by:

1. Defining and storing protection parameters in a Sentinel HASP Envelope configuration file
2. Accessing the Linux project file during the protection session

Note:

Sentinel HASP Envelope can be used on a 32-bit or 64-bit platform to protect both 32-bit and 64-bit executables and shared objects.

Sentinel HASP Envelope configuration parameters are detailed in *Envelope Configuration Settings for Linux.pdf*, available in the **Docs** folder on the Sentinel HASP installation DVD.

Activating Sentinel HASP Envelope on Linux

After protection parameters are defined and stored in the Sentinel HASP Envelope configuration file, you activate Sentinel HASP Envelope using one of the following methods:

- Using the command-line utility, specify the mandatory Sentinel HASP Envelope parameters, then run Sentinel HASP Envelope using the default settings.
- Open the Sentinel HASP Envelope configuration file and edit the settings as required. Using the command-line utility, program the utility to parse the Sentinel HASP Envelope configuration file, then run Sentinel HASP Envelope. The configuration file is called `envconfig.cfgx`

Note:

When running the Sentinel HASP Envelope command-line utility, use either the command-line switches, or the configuration file.

Overwrite Parameters for Configuration File

Several parameters defined in the configuration file can be overwritten by the optional arguments listed in the following table.

Parameter	Description
<code>-v --vcf:<filename></code>	Vendor Code file
<code>-f --fid:<id></code>	Feature ID. If no Feature ID is specified, default Feature ID will be used.
<code>-b --bgchk:<time></code>	Enables background checks. Time is in seconds. (0 = off, default = 300)
<code>-e --enclevel:<level></code>	Encryption level. (1–5, default = 4)
<code>-c --cfg:<file></code>	Sentinel HASP Envelope configuration file for Linux
<code>-m --msg:<file></code>	Message file
<code>-m --msg-out:<val></code>	The message output mode at run-time. 0 = no message output 1 = GUI 4 = console 5 = GUI and console (default = 1)
<code>-m --wchar</code>	Writes run-time errors as wide character strings. Required when you specify that messages will be output to a console (values 4 or 5 in the previous switch).
<code>-h --help</code>	Displays help screen
<code>-q --quiet</code>	Specifies that only error and warning messages are displayed

Sentinel HASP Envelope for Mac Binaries

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

Sentinel HASP Envelope Prerequisites for Mac

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

- Sentinel HASP Run-time Environment
- Sentinel HASP Vendor Suite, containing the Sentinel HASP Envelope and the MasterHASP wizard
- A valid Vendor Code stored in the **VendorCodes** folder.
For additional information, see *Extracting the Vendor Code from Sentinel HASP Vendor Keys* on page 60.
- The Mac binaries that you want to protect

Running Sentinel HASP Envelope for Mac

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

To access the command-line version of Sentinel HASP Envelope, go to:
...\\MacOS\\VendorTools\\VendorSuite\\envelope_darwin

Type **envelope_darwin -h** in the command line to start the command-line version of Sentinel HASP Envelope.

Sentinel HASP Envelope for Mac Protection Parameters

After your Mac executable or dynamic library has been included in a Sentinel HASP project, protection can be performed effortlessly, based on the default Sentinel HASP 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 HASP 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 HASP Envelope Help documentation.

Sentinel HASP Envelope for Java Executables

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

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 HASP Envelope.

Java Considerations

When protecting Java executables, consider the following issues:

- The methods selected for protection by Sentinel HASP 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 HASP Envelope Help documentation.
- Sentinel HASP 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 program may cause a deadlock when it executes a protected paint method at runtime with no HASP 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 HASP 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 protection, 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 HASP Envelope Prerequisites for Java

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

- The Java JRE or JDK must be installed
- Sentinel HASP Run-time Environment
- Sentinel HASP Vendor Suite, containing the Sentinel HASP Envelope and the MasterHASP wizard

- A valid Vendor Code stored in the **VendorCodes** folder.
For additional information, see [Extracting the Vendor Code from Sentinel HASP Vendor Keys](#) on page 60.
- The JAR or WAR executables that you want to protect

Before your JAR/WAR archive is protected, include the following customized Run-time API dynamic libraries with the archive:

Operating System	Customized Run-time API Dynamic Libraries
Windows (32/64-bit)	hasp_windows_****_<vendorId>.dll
Mac OSX	hasp_darwin_<vendorId>.dylib
Linux (32/64-bit)	libhasp_linux_***_<vendorId>.so

During protection of the Java applications, Envelope copies these libraries automatically to the output directory.

For your protected Java executables to function at run-time, one or more HASP DLLs are required. For more information, see [“Protection-related Software”](#) on page 208

Running Sentinel HASP Envelope for Java Engines

In the **Start** menu, select **SafeNet > Sentinel HASP > Vendor Suite**. From the Sentinel HASP Vendor Suite program selection screen, launch Sentinel HASP Envelope.

Sentinel HASP Envelope for Java Protection Parameters

After your Java executable has been included in a Sentinel HASP project, protection can be performed, starting from the default Sentinel HASP 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 HASP 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 HASP Envelope, you can determine protection attributes and aspects of the behavior of the protected program.

Protected Program Behavior

Sentinel HASP 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 HASP protection key.

All parameters and procedures are detailed in the Sentinel HASP Envelope Help documentation.

Protection Strategies

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

In this chapter:

- Overview
- General Protection Guidelines
- Types of Attack and Their Sentinel HASP Defense

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 chapter 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 HASP measures that you can use to enhance your software security.

In addition to the information described in this manual, our team of SafeNet 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 SafeNet, contact your local SafeNet representative.

General Protection Guidelines

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

SafeNet thoroughly and constantly investigates potential and actual threats to software security, and Sentinel HASP 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 HASP protection software that is available. The latest software can be downloaded from the Sentinel website, at <http://www3.safenet-inc.com/support/hasp-srm/vendor.aspx>.

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 Cracking Attempt is Detected

When a cracking attempt is detected (for example, through using a checksum—described later in the chapter), 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 HASP 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 HASP 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 HASP Solution

The more files that are protected, the longer it takes a software cracker to remove protection. You can protect multiple executable and DLL files using Sentinel HASP Envelope. You can also use the DataHASP feature of Sentinel HASP Envelope to encrypt and protect data files.

Modifying Key Memory

Licensing data is normally stored in the memory of a software protection key. A software cracker attempts to access the 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 HASP Solution

In the context of Sentinel HASP, Read-only memory (ROM) is a segment of the memory that can contain data that the protection application can access, but cannot overwrite. Sentinel HASP keys contain two ROM segments, one of which contains Sentinel HASP 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 HASP automatic Feature-based licenses utilize read-only memory of Sentinel HASP protection keys. The variety 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 HASP 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, *Sentinel HASP Licensing Models*.

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 attempt to replace the driver.

Sentinel HASP Solution

Sentinel HASP provides a secure channel between an application and the HASP 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 HASP 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 Terminal Servers and Terminal Service Solutions

When using the terminal servers of some operating systems, it might be possible for an end user with a locally connected protection key to enable software on multiple concurrent terminals.

Sentinel HASP Solution

The Sentinel HASP protection includes mechanisms to determine if a protected application is running on a terminal server. If such a scenario is detected, and the license is on a local protection key, the program will not function, subject to vendor overrides. If the license is on a network key, one of the concurrent licenses will be consumed, subject to vendor overrides.

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 HASP Solution

HASP HL keys are each unique and have their own ID. Keys that are in the same batch 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 HASP HL key is copied to another HASP 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 HASP Solution

Use either the HASP HL Time or HASP HL NetTime keys when implementing time-based licenses for your software. Neither the clock itself, or the license which is stored in read-only memory, can be modified.

Additional Sentinel HASP-specific Strategies

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

Use Both the Sentinel HASP Run-time API and Sentinel HASP Envelope

Maximize security by using the Sentinel HASP Run-time API to implement calls to a Sentinel HASP protection key, and protect the application with Sentinel HASP 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 HASP 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 HASP Protection Key

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

Encrypting data with the Sentinel HASP 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 HASP protection key and its internal intelligence.

By implementing a Sentinel HASP Run-time API scheme in which data is decrypted by a Sentinel HASP protection key, the association between the protected application and the Sentinel HASP 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 HASP 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.

Working with the DataHASP Encryption Utility

The chapter describes data file protection using the DataHASP utility.

In this chapter:

- Introduction
- DataHASP Prerequisites

Introduction

The Sentinel HASP Envelope DataHASP encryption utility enhances the default Sentinel HASP Envelope protection by injecting a protected program with the ability to encrypt and decrypt specified data files. Decryption and encryption can be applied to files that have been pre-encrypted, and to new files created by your protected application. Decryption occurs when a data file is opened and encryption occurs when a data file is saved.

The ability to implement DataHASP functionality is enabled when the executable file is being protected using Sentinel HASP Envelope. The types of files that are to be encrypted and decrypted are also specified at the same time.

Note:

It is important that you use DataHASP to pre-encrypt any data files that you intend to distribute with your protected program.

When to Encrypt 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 represent your investments, so it is worthwhile preventing your intellectual property from being exposed without protection.

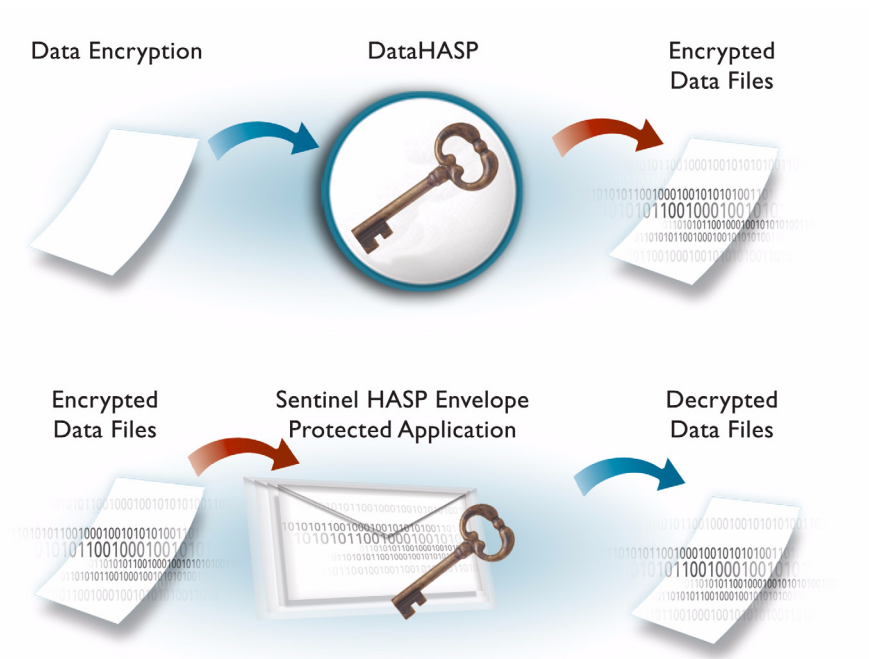
DataHASP Users

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

Sentinel HASP Data File Handling

DataHASP generates data files that can be processed by programs protected by Sentinel HASP Envelope. After a specific set of data files has been encrypted, those files can be accessed by Sentinel HASP Envelope.

The following figure illustrates the actions related to encrypting data files that take place between Sentinel HASP Envelope and DataHASP. After data files are encrypted, those files can only be accessed and decrypted by applications protected with Sentinel HASP Envelope.



Operative Interaction between Sentinel HASP Envelope and DataHASP

DataHASP Prerequisites

To use DataHASP, you must have the following components installed. In addition, you need to know the location of the appropriate Sentinel HASP Envelope project, and the files that you want to encrypt.

- **Sentinel HASP protection keys:** Ensure that the Sentinel HASP protection key that is used to protect the program is accessible when protecting data.

While encryption and decryption of data is performed in the protected device driver, the Sentinel HASP protection key generates the encryption key. This method ensures maximum protection while maintaining the high performance that is required when large volumes of data are decrypted.

- **Run-time Environment:** Before using the DataHASP utility, ensure that the appropriate run-time environment is installed on your machine.
- **Input:** To use DataHASP correctly, the following must be readily available in known directories:
 - ◆ A saved Sentinel HASP Envelope project containing a program or programs with enabled data filters
 - ◆ Data files you want to encrypt. These files must satisfy the file filters specified for the program(s)

Launching DataHASP

You can launch DataHASP directly from Sentinel HASP Envelope after you have defined data filters. Alternatively, you can click the `datahasp.exe` file, located in the following directory on your system:

...\Program Files\SafeNet\Sentinel HASP\VendorTools\VendorSuite

Supported Functionality

The DataHASP utility has a conveniently designed interface that manages all aspects of data encryption projects. The interface enables you to:

- Open Sentinel HASP Envelope projects and list the programs they contain
- Edit existing DataHASP projects
- Add files or directories for data encryption
- Encrypt data files and directories for specific programs protected by Sentinel HASP Envelope
- View and save encryption process logs
- Access all Sentinel HASP Vendor Suite applications and the Sentinel website

Modifying Input

Data files are subject to regular modification. The design of DataHASP anticipates the requirement to regularly update data file content. After you have encrypted modified data files, they can only be accessed if a specific Sentinel HASP protection key is detected.

Part 3 Licensing

In this section:

- **Chapter 7: Introduction to Sentinel HASP Business Studio**
Provides an overview of Sentinel HASP Business Studio and the major processes it facilitates, lists its prerequisites, and explains how to use the application.
- **Chapter 8: Preparing Your Sentinel HASP Licensing Plan**
Outlines the importance of licensing your software products, describes the licensing options provided by Sentinel HASP, and explains how to prepare a licensing plan for use with Sentinel HASP Business Studio.
- **Chapter 9: Implementing Your Sentinel HASP Licensing Plan**
Describes how to use Sentinel HASP Business Studio to define and manage the Features and Products included in your Sentinel HASP licensing plan, and how to maintain Products and licenses as circumstances change.
- **Chapter 10: Sentinel HASP Orders, Production, and Development Tasks**
Describes how to use Sentinel HASP Business Studio to manage and produce orders, and to perform additional development-related tasks.
- **Chapter 11: Sentinel HASP Administration and Customer Services**
Describes how to use Sentinel HASP Business Studio to define Sentinel HASP user details, maintain Batch Codes, configure system settings, perform manual Product activation and maintain customer data.
- **Chapter 12: Sentinel HASP Remote Update System**
Describes the Sentinel HASP Remote Update System (RUS) utility and explains how to use RUS to remotely update license data in deployed Sentinel HASP protection keys.

- **Chapter 13: Generating Sentinel HASP Reports**

Provides an overview of the Sentinel HASP Reporting facility and describes some of the main features of the facility.

Introduction to Sentinel HASP Business Studio

This chapter provides an overview of Sentinel HASP Business Studio and the major processes it facilitates. It also describes the user roles and their functions in Sentinel HASP Business Studio, lists its prerequisites, and explains how to start using the application.

In this chapter:

- [Sentinel HASP Business Studio Overview](#)
- [Sentinel HASP Business Studio Users and User Roles](#)
- [Distribution Channels](#)
- [Getting Started With Sentinel HASP Business Studio](#)

Note:

This chapter provides high-level information on Sentinel HASP Business Studio processes. For detailed practical instructions for using each function in Sentinel HASP Business Studio, see the Sentinel HASP Business Studio Help documentation.

Sentinel HASP Business Studio Overview

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

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

Sentinel HASP Business Studio Major Workflows

Sentinel HASP Business Studio contains two major components—the Business Studio client application and the Business Studio Server. Together they handle three major workflows: license planning, order processing and production, and software activation.

License Planning

Before starting to use Sentinel HASP Business Studio, 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 HASP, 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 HASP Business Studio.

In addition, the licensing plan can include the *Products* that your company wants to sell and/or distribute for evaluation. In Sentinel HASP, 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 HASP Business Studio. The output of this process is a repository of Products that are stored in the Sentinel HASP database—ready for customer orders.

Note:

You can make subsequent 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 HASP, see Chapter 8, *Preparing Your Sentinel HASP Licensing Plan*.

For a description of the many types of model licenses you can implement using Sentinel HASP, see Part 5, “Sentinel HASP Licensing Models” on page 233.

For additional information on defining Features and Products in Sentinel HASP Business Studio, see Chapter 9, *Implementing Your Sentinel HASP Licensing Plan*.

Order Processing and Production

Staff in your organization’s orders department receive and fulfil orders. An *order* is a request for Sentinel HASP items, and can be one of the following:

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

Order processing personnel process the order details using Sentinel HASP Business Studio. The license terms of each Feature in the ordered Products may be specified when the Product is defined, or when the order is processed.

When all the details of an order have been defined, the order can be produced. The Product details, including the license terms and memory data, are stored in the specified Sentinel HASP 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 orders in Sentinel HASP Business Studio, see Chapter 10, *Sentinel HASP Orders, Production, and Development Tasks*.

Software Activation and Online Updates

Product activation and *online* updates are performed by means of the Sentinel HASP Business Studio Server when your software is at the end user's site.

Product Activation with HASP SL Keys

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

1. A Product Key is produced in Sentinel HASP Business Studio and supplied to the end user.
2. The end user sends the Product Key to the Sentinel HASP Business Studio Server for validation.
3. A HASP 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 HASP Update information is stored on the Sentinel HASP Business Studio Server for use in software that you provide to your end users. The update is then implemented as part of the end users' installation process.
- A file that contains the HASP Update information is generated and sent to the end user. This file can then be used with the Sentinel HASP Remote Update System (RUS) utility to ensure secure, remote updating of the deployed Sentinel HASP protection keys.

For additional information on RUS, see Chapter 12, *Sentinel HASP Remote Update System*.

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

Sentinel HASP Business Studio Users and User Roles

Sentinel HASP Business Studio 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 HASP Administrator.

The following user types are available.

- **Standard user** — can be assigned any roles and can access all data for which those roles are authorized.
- **Distribution Channel user** — limited to certain roles and to a specific subset of the data for which those roles are authorized. For more information, see [Distribution Channels on page 123](#).

The table that follows describes the roles available. (Almost all of the tasks listed in the table relate to functionality in Sentinel HASP Business Studio.)

Role	Authorized Tasks	For more information
Product Management	<ul style="list-style-type: none"> Define and manage Features and Products 	Chapter 8— <i>Preparing Your Sentinel HASP Licensing Plan</i> Chapter 9— <i>Implementing Your Sentinel HASP Licensing Plan</i>
Order Management	<ul style="list-style-type: none"> Define and manage customers Enter and manage orders 	Chapter 10— <i>Sentinel HASP Orders, Production, and Development Tasks</i>
Production	<ul style="list-style-type: none"> Produce orders 	Chapter 10— <i>Sentinel HASP Orders, Production, and Development Tasks</i>
Customer Services	<ul style="list-style-type: none"> Define and manage customers Manage Product activations 	Chapter 11— <i>Sentinel HASP Administration and Customer Services</i>
Report Generation	<ul style="list-style-type: none"> Run and view reports Schedule generation of and arrange distribution of reports. 	Chapter 13— <i>Generating Sentinel HASP Reports</i>
Development	<ul style="list-style-type: none"> Perform development-related tasks Operate Sentinel HASP Toolbox and Sentinel HASP Envelope 	Chapter 10— <i>Sentinel HASP Orders, Production, and Development Tasks</i>
Administration	<ul style="list-style-type: none"> Manage Sentinel HASP users Maintain Master keys Configure system settings Manage Distribution Channels Able to perform Report Generation role tasks 	Chapter 11— <i>Sentinel HASP Administration and Customer Services</i> <i>Distribution Channels</i> (below)

Note: The “HASP” user is authorized to perform all functions in Sentinel HASP.

Distribution Channels

Many vendors sell their products through distributors. Distributors typically require access to Sentinel HASP Business Studio so that they can enter and produce orders. However, in many cases a vendor may want to define Products differently for different distributors due to regional or other considerations. Furthermore, the vendor may not want to allow each distributor to access the orders entered by other distributors.

Sentinel HASP Business Studio enables you to create an insulated *Distribution Channel* for each distributor. The Business Studio users assigned to a given Distribution Channel:

- can only enter and produce orders for Products associated with that Distribution Channel.
- can only access orders associated with the Distribution Channel.

This gives the vendor the freedom to define different business models for different distributors without duplication of effort in defining Products.

The Distribution Channel mechanism works as follows:

- The Administrator defines one or more Distribution Channels in Sentinel HASP Business Studio.
- Each Product that can be ordered and produced by a specific Distribution Channel is associated by the Administrator with that Distribution Channel. (A Product can be associated with multiple Distribution Channels.)
- When defining user accounts for each distributor, the Administrator defines each user as a *Distribution Channel user* and assigns the user to a specific Distribution Channel. A Distribution Channel user can only be assigned the roles **Order Management** or **Production**.
- When a Distribution Channel user logs in to Sentinel HASP Business Studio, the user can only:
 - ◆ enter orders for Products associated with the user's Distribution Channel.
 - ◆ view or produce orders that were entered by a user assigned to the same Distribution Channel.

The use of Distribution Channels is optional and can be disabled by an Administrator.

A *Standard user* (one who is not assigned to any Distribution Channel) is not restricted from accessing Products or orders that are associated with a Distribution Channel.

Note:

Once an order has been entered by a Distribution Channel user, a Standard user cannot add a Product to the order unless the Product is associated with the same Distribution Channel as the original order.

This restriction does not apply if the original order was entered by a Standard user.

Getting Started With Sentinel HASP Business Studio

Before you start to use Sentinel HASP Business Studio, ensure that:

- Sentinel HASP Vendor Suite is installed on your machine
- You have received a Sentinel HASP user name and password from your Sentinel HASP system administrator

After you have logged in to Sentinel HASP Business Studio, you can change the Sentinel HASP password that you received to a password of your own choice. For additional information on changing your password, see the Sentinel HASP Business Studio Help documentation.

Note:

Sentinel HASP passwords are case-sensitive, so ensure that you use upper-case and lower-case letters correctly when you type your password.

Prerequisites for the Sentinel HASP Administrator

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

- A valid connection to the Sentinel HASP Business Studio Server must exist. For additional information on installing the Sentinel HASP Business Studio Server, see the *Sentinel HASP Installation Guide*.
- You must have a Sentinel HASP Master key that contains your license for Sentinel HASP and your company's specific Vendor Code. If not previously introduced, the Sentinel HASP Master key is introduced during the Sentinel HASP Business Studio Server installation process.
- The Sentinel HASP Master key must remain connected to the Sentinel HASP Business Studio Server machine in order to enable you to perform Sentinel HASP Business Studio functions. If the Sentinel HASP Business Studio Server is installed on more than one server machine, each server must have a separate Sentinel HASP Master key locally connected.

Note:

If you are evaluating Sentinel HASP Business Studio, you can use the **DEMOMA** Batch Code provided, which does not require a Sentinel HASP Master key.

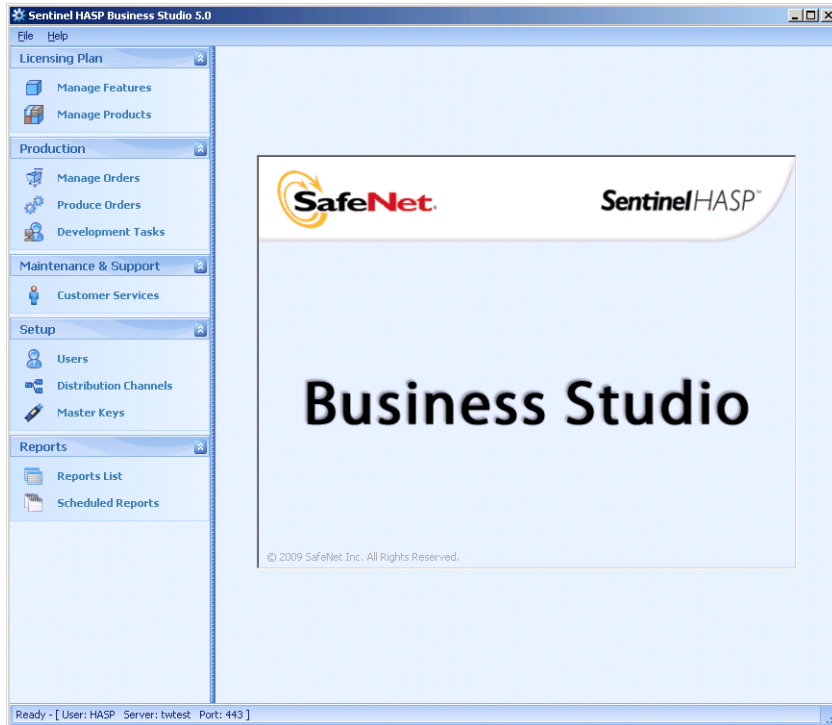
- You must define user names, passwords, roles, and batch access for each Sentinel HASP Business Studio user, and also for yourself. For additional information, see [Maintaining User Details on page 187](#).

A default user name and password is provided with Sentinel HASP to enable you to log in to Sentinel HASP Business Studio as the Sentinel HASP Administrator. The default user name and password is **HASP**.

For additional information on the Sentinel HASP administration tasks and options in Sentinel HASP Business Studio, see [Administration Tasks on page 186](#).

Sentinel HASP Business Studio Window

When you log in, the Sentinel HASP Business Studio launch screen is displayed in the Main pane of the Sentinel HASP Business Studio window.



On the left is the Sentinel HASP Business Studio Function pane. This pane displays the function groups available for your use, depending on the user roles assigned to you.

When you click one of the functions in the Function pane, the window for that function is displayed in the Main pane. This window contains:

- Information relevant to the function you selected
- A Task pane on the right side of the window, containing tasks that you can perform within the selected function

For example, if you click **Manage Features**, the Manage Features window is displayed. Details of currently defined Features are displayed in the Main pane. The Task pane lists the tasks that can be performed, such as defining a new Feature, opening an existing Feature, deleting a Feature, and so on.

Many of the function windows provide filter and/or search fields to enable you to quickly locate required data.

Using the Sentinel HASP Business Studio Help

Detailed instructions for using each function and task in Sentinel HASP Business Studio are provided in the Sentinel HASP Business Studio Help documentation.

To open the Sentinel HASP Business Studio Help:

- From the **Help** menu in the Sentinel HASP Business Studio window, select **Sentinel HASP Business Studio Help**. The Sentinel HASP Business Studio Help documentation is displayed.
- To open context-sensitive Help for the current task or function, press the **F1** key or click **Help** in any Sentinel HASP Business Studio window or dialog box. The Help topic associated with the current task or function is displayed.

Preparing Your Sentinel HASP Licensing Plan

Before you start to use Sentinel HASP Business Studio in your organization, you may want to prepare a detailed licensing plan for use with Sentinel HASP. Although it is recommended that you prepare a licensing plan, it is not a prerequisite for using Sentinel HASP Business Studio. Licensing decisions can be implemented or varied on-the-fly.

This chapter outlines the importance of licensing your software products, describes the licensing options provided by Sentinel HASP, and suggests how you might prepare a detailed licensing plan for use with Sentinel HASP Business Studio.

In this chapter:

- Licensing Overview
- Preparing Your Licensing Plan
- Choosing the Protection Level for Your Products
- Designating Products for Trial or Grace Period Use
- Assigning License Terms to Features
- Utilizing HASP Memory
- Using Your Licensing Plan With Sentinel HASP Business Studio

Note:

This chapter provides high-level information about Sentinel HASP licensing options. For detailed practical instructions for implementing the licensing options in Sentinel HASP Business Studio, see the Sentinel HASP Business Studio Help documentation.

Licensing Overview

Part 2, *Protection*, 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 fully-featured version.

Sentinel HASP's versatility 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, [Sentinel HASP Licensing Models](#).

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 models required to meet your company's business needs.

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

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

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

You can define additional licensing 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 HASP to streamline the implementation of your company's licensing strategy. Your Sentinel HASP 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 HASP 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 HASP 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 HASP Business Studio.

Note:

You can make subsequent changes to your licensing plan and license 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 HASP, 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 HASP, 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 HASP, 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 HASP, 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

Note:

The REPORT GENERATOR Feature has not yet been fully developed and is not currently included in the **HQ Design Pro** Product. This Feature is planned for a future release.

Choosing the Protection Level for Your Products

Your choice of the Sentinel HASP 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 HASP protection keys are available:

- **HASP HL keys:** The hardware-based protection and licensing component of Sentinel HASP that provides the safest and strongest level of protection.
- **HASP SL keys:** The software-based protection and licensing component of Sentinel HASP—virtual HASP HL keys.

(For additional information, see [HASP SL Keys](#) on page 41.)

Your software and the user license are both *locked* to the Sentinel HASP protection key that you select.

When you define the Products for inclusion in your licensing plan, you also select which Sentinel HASP *locking type* to assign to each Product. The locking type determines the level of protection for each Product, as follows:

- **HASP HL locking only:** hardware-based level of protection
- **HASP SL locking only:** software-based level of protection
- **HASP HL or HASP SL locking:** software-based level of protection

HASP HL Key Protection and Activation

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

Benefits of HASP HL Key Protection

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

In addition, HASP HL key protection:

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

HASP SL Key Protection and Activation

A Product that is protected with a HASP SL key can be activated only after the following steps have been completed:

1. A *Product Key*, consisting of a string of characters, is generated in Sentinel HASP Business Studio and supplied to the end user.
2. The end user returns the Product Key as proof of purchase.

3. The Product Key is sent to the Sentinel HASP Business Studio Server for verification.
4. A HASP SL key with license terms is sent back and installed on the end user's computer.

Benefits of HASP SL Key Protection

With HASP SL key protection:

- 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 HASP SL key, you can specify that a license can be detached from the pool of network seats and attached to a remote recipient machine.

Specifying the Protection Level for Individual Orders

Sentinel HASP gives you the flexibility to choose the Sentinel HASP 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 **HASP HL or HASP SL** locking type to a Product. With this locking type, the decision on which type of Sentinel HASP protection key is to be shipped with the Product is made when each order is processed.

A Product that is assigned the **HASP HL or HASP SL** locking type can be sent to the end user with either a HASP HL key or a HASP SL key.

WARNING!

A Product that can be distributed with both HASP HL keys and HASP SL keys is always supplied with the HASP SL key-level of protection, even when it is shipped with HASP HL keys.

Designating Products for Trial or Grace Period Use

Sentinel HASP 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 of into a fully-functional version by activating it with a Sentinel HASP protection key.

You can also use Sentinel HASP 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 HASP protection key.

Sentinel HASP enables you to define trial and grace periods for software protected with any type of Sentinel HASP protection key.

For example, software protected with HASP HL keys can be purchased and delivered over the Internet while the HASP HL keys are shipped, and end users can start using the software while waiting for the arrival of their key.

Similarly, end users who purchase and install a software application can use it for a 30-day grace period without activating it. During this grace period, they can activate the software remotely and receive a HASP 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, it will stop running until activated by the end user.

In Sentinel HASP, a Product that is intended for distribution as trialware or for use during a grace period is referred to as a *Provisional Product*. Sentinel HASP locking types are not applicable to Provisional Products, since these Products are distributed without Sentinel HASP protection keys.

Your licensing plan can include all the Provisional Products to be offered by your organization.

Assigning License Terms to Features

Sentinel HASP 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 type*:

- **Perpetual:** Indicates that the license can be used an unlimited number of times for an unlimited period of time.
- **Expiration Date:** Specifies the date on which the license expires.
- **Executions:** Specifies the maximum number of times that the Feature can be used.
- **Time Period:** Specifies the number of days until the license expires, from the date of first use.

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. In addition, if the Feature will be used in Products that are locked to HASP 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 HASP 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 type or the number of concurrent instances for each Feature in the Product. When each order for the Product is processed, the person processing the order defines 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, fully-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 a Provisional Product, to enable it to be distributed freely for evaluation
- **HQ Design Lite** is supplied with HASP SL key protection, enabling electronic distribution
- **HQ Design Pro** is supplied with HASP 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:	<i>Provisional</i>	<i>HASP SL keys</i>	<i>HASP HL keys</i>
License 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 HASP Memory

All Sentinel HASP protection keys—with the exception of HASP 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 the memory, 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 HASP memory of the selected Sentinel HASP 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.

Using Your Licensing Plan With Sentinel HASP Business Studio

Your licensing plan can be implemented using Sentinel HASP Business Studio. As your licensing requirements change, you can revise the licensing plan and ensure that the changes are implemented using Sentinel HASP Business Studio. 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 Chapter 9, *Implementing Your Sentinel HASP Licensing Plan*.

Sentinel HASP 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.

Implementing Your Sentinel HASP Licensing Plan

This chapter is intended for Sentinel HASP Business Studio users who are assigned the **Product Management** role. It describes how to use Sentinel HASP Business Studio to define and manage Features and Products in Sentinel HASP, and to maintain Products and licenses as circumstances change.

For information on preparing a licensing plan and on Sentinel HASP licensing options, see Chapter 8, *Preparing Your Sentinel HASP Licensing Plan*.

For an overview of Sentinel HASP Business Studio and for information on starting to use the application, see Chapter 7, *Introduction to Sentinel HASP Business Studio*.

In this chapter:

- License Planning in Sentinel HASP Business Studio
- Managing Features
- Managing Products
- Maintaining Products and Licenses

Note:

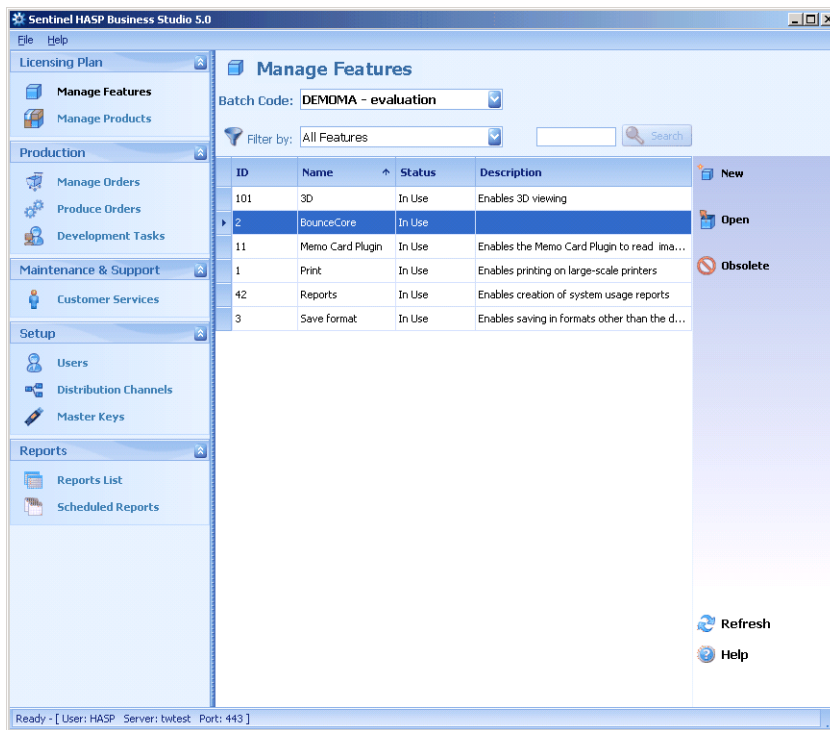
This chapter provides high-level information on license planning and definition processes. For detailed practical instructions for using each function in Business Studio, see the Sentinel HASP Business Studio Help documentation.

License Planning in Sentinel HASP Business Studio

Before you start to use Sentinel HASP Business Studio for license planning, it is suggested that you prepare a licensing plan. For additional information, see Chapter 8, *Preparing Your Sentinel HASP Licensing Plan*.

When you start Sentinel HASP Business Studio, you have access to the Licensing Plan group of functions, including:

- Managing Features
- Managing Products



Each of these function is described in this section.

Note:

All Sentinel HASP Features and Products are associated with a Sentinel HASP Batch Code. For additional information on Batch Code, see [Personalized Vendor and Batch Codes on page 39](#).

Managing Features

When you select the **Manage Features** function in the Sentinel HASP Business Studio window, you can view the details of all defined Features associated with the selected Batch Code. You can perform the following tasks using the **Manage Features** function in Sentinel HASP Business Studio:

- Define Features
- Withdraw Features from use

Defining Features

If you have prepared a licensing plan, the first stage in its implementation is to use Sentinel HASP Business Studio 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:

- The Batch Code associated with the Feature
- A Feature Name that is unique in the selected batch (mandatory). The maximum length for a Feature Name is 50 characters.
- A free-text description that provides additional information about the Feature (optional)
- The ID number that you want to assign to the Feature (optional). The ID must be unique in the selected batch. The same Feature ID may be used in more than one batch.

After you have defined a Feature, and until the Feature is included in a Product, you can change these properties in Sentinel HASP Business Studio. After the Feature has been included in one or more Products, you can open the Feature to view its details, but you cannot change them.

Note:

License terms are Feature-specific in Sentinel HASP. However, they are not defined as part of the Feature properties. The license terms for a Feature are specified when the Feature is added to a Product, or when the Product is added to an order. 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 of the order.

Feature Identification

By default, Sentinel HASP Business Studio generates a unique Feature ID for each new Feature. You can assign your own numeric identifier to the Feature, for example, to maintain consistency with existing Feature data. The Feature ID that you specify must be unique in the selected batch.

Transferring Feature Definitions for Development Use

After you have defined the Features for a selected batch, 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 182](#).

Feature Status Values

When a Feature is first defined, its status is **Ready**. This status indicates that the Feature is available for use in Products. The Feature details can be changed until it is included in a Product, and if required, the Feature can be deleted.

After a Feature has been selected for use in a Product, its status changes to **In use**. This status indicates that the Feature is available for use in additional Products, but the Feature details cannot be changed and it cannot be deleted.

If a Feature has been used in at least one Product and is subsequently withdrawn from use, its status changes to **Obsolete**. You can restore a Feature that was previously made obsolete.

Withdrawing a Feature

At some stage, you may want to withdraw a selected Feature from use and specify that it can no longer be included in Products, for example, if the functional component associated with the Feature is being replaced by a more sophisticated component.

If the Feature has not been included in any Product, you can delete it. A Feature cannot be deleted once it has been included in at least one Product. You can, however, withdraw the Feature from use by marking it as **Obsolete**.

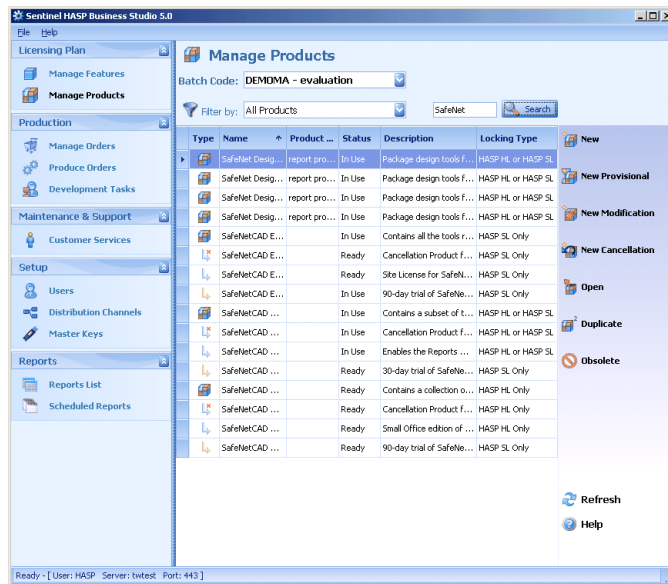
An Obsolete Feature cannot be added to Products, but its details are maintained in Sentinel HASP Business Studio for tracking purposes, and it continues to be functional in existing Products.

Restoring a Feature

An Obsolete Feature can be restored. A restored Feature can be added to Products and can be used in the same way as any other Feature.

Managing Products

When you select the **Manage Products** function in the Sentinel HASP Business Studio window, you can view the details of all defined Products associated with the selected Batch Code.



You can perform the following tasks using the **Manage Products** function in Sentinel HASP Business Studio:

- Define new Base Products
- Define new Provisional Products
- Duplicate 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 or delete a Product that has already been included in an order (with the **In Use** status).

Defining New Products

Before you start to define the new Products in your licensing plan, ensure that you have the following information available for each Product:

- The Batch Code associated with the Product
- A Product Name that identifies the Product and is unique in the selected batch (mandatory). The maximum length for a Product Name is 50 characters.
- A free-text description that provides additional information about the Product, for example, the functionality it includes (optional)
- Product reference information that can identify the Product in a different system, for example, a product code in your company's ERP system (optional)
- The level of protection (locking type) that you want to apply to the Product
- The Features to be included in the Product
- The license terms for each Feature to be included in the Product
- The data to be stored in the memory associated with the Product

After a Product has been defined, it can be included in orders. For additional information on processing orders, see [Defining Orders on page 166](#).

Until the Product is included in an order, you can change the Product properties, Features, and memory contents in Sentinel HASP Business Studio. After the Product has been included in at least one order, you can open the Product to view its details. However, you cannot make any changes.

The only changes that can be made after a Product is included in an order are those related to licensing terms and memory data that have been previously specified as definable at order time, and these changes are made when the order is being processed.

Product Types

The basic unit on which all Products are built is the *Base Product*. A Base Product can contain all the Product attributes such as Features, licensing data and memory—and can be used as a Product that you offer for sale, and/or as a “shell” on which other Product types are built.

You can define *Provisional Products* for use during a grace period or as trialware. The properties for Provisional Products are not identical to those for standard Products. For additional information, see [Defining Provisional Products on page 155](#).

You can create *Duplicate Products*, which in effect “copy and paste” existing Product details into a new Product. For additional information, see [Duplicating a Product on page 156](#).

You can also define *Modification Products* and *Cancellation Products* to cater for changes in your Product range and in your customers’ requirements. For additional information, see [Maintaining Products and Licenses on page 156](#).

Selecting the Locking Type for the Product

When you define a Product, you must select a locking type. The locking type determines:

- The level of protection for the Product
- The type of Sentinel HASP protection keys that can be shipped with the Product
- The way that the Product can be activated

The locking type options are:

- **HASP HL locking only:** The Product receives the highest level of protection and can be shipped and activated with HASP HL keys only. For additional information on HASP HL locking, see [*HASP HL Key Protection and Activation on page 135*](#).
- **HASP SL locking only:** The Product can be shipped and activated with HASP SL keys only. For additional information on HASP SL locking, see [*HASP SL Key Protection and Activation on page 135*](#).
- **HASP HL or HASP SL locking:** The decision on the type of Sentinel HASP protection key to be shipped with the Product is made when each order is processed.

WARNING!

A Product that can be distributed with both HASP HL keys and HASP SL keys is always supplied with the HASP SL key-level of protection, even when it is shipped with HASP HL keys.

The default locking type can be specified using the System Settings function.

Protection Against Cloning

This section describes the options available for protecting your software against cloning of the physical or virtual machine on which the software is installed.

One of the methods sometimes employed to enable the illegitimate 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.

With Sentinel HASP, you have the ability to detect probable machine cloning and to disable clone-protected software that is locked to HASP SL keys. Clone detection is effective whether the protected software is installed on a physical machine or on a virtual machine.

Note:

When software is locked to a HASP 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 HASP HL key to which the software license is locked.

You specify whether you want clone protection to be applied to a Product at the time that you define the Product's properties using Business Studio. Clone protection is selected by default when you specify that a Product can be locked to a HASP 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. (For information on how Sentinel HASP detects cloned software, see Appendix D, *How Sentinel HASP Detects Machine Cloning*.)

When the Sentinel HASP Run-time 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 HASP License Manager and displayed in the Sentinel HASP Admin Control Center. For additional information, refer to the online help documentation in the Admin Control Center.

The following workflows provides an overview of how to enable clone detection for licenses locked to HASP SL keys, and how to manage licenses that have been disabled due to the detection of machine cloning.

During software protection:

- During protection of your software, use the Run-time 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 HASP 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 Business Studio:
 - ◆ Ensure that **Clone Protection** is enabled (by default, this is enabled).
 - ◆ 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:

1. Following contact by an end user, begin the software activation service in Business Studio. When Business Studio detects cloning via the C2V file, it displays a dialog box to enable you to generate a V2C file that will clear the disabled software license on the end-user machine. This process can only be implemented manually.

Note:

Clearing a HASP SL key does not reformat the key, it only removes license terms of Products that were locked to the key. Clearing does not remove data from the key's memory, and it does not remove Provisional Products from the key. The HASP SL key is given a new key ID.

2. Send the V2C file to the end user to apply to the machine.
The end user applies the V2C file to the machine on which cloning was detected. The license for the software that was disabled is cleared and a Provisional Product is enabled. Any consumption of the original Provisional Product is retained. For example, if the Provisional Product provided 90 days of trial use and 30 days have already been used, only 60 days will be still available.
3. To fully activate the software, the end user must initiate the software activation process, in the normal way.

For additional information about enabling a HASP SL key that was disabled due to clone detection, see the *Sentinel HASP Business Studio* online help.

Additional Information about Clone Protection

- If you attempt to check in a C2V file, and Business Studio detects that the C2V is from a cloned machine, you cannot check the file into the Sentinel HASP database. Similarly, you cannot use a C2V file from a cloned machine to create a license update.
You can click **C2V Details** in the Check in C2V and Key Data dialog box, to view details of the C2V if required.
- You can enable and disable clone protection by creating a Modification Product. A Modification Product must include new or modified Features. If you change the clone protection setting, it also changes the Base Product from which the Modification Product was built, and all other Products created from that Base Product.

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
 - ◆ Executions
 - ◆ Time Period
- Assign a value for the selected license type:
 - ◆ 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:** Each login request for a single machine is counted as an instance (default)
- **Login:** Each login request is counted as an instance
- **Process:** Each login request for a single process is counted as an instance

If the Feature is in a Product that will be locked to a HASP 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 HASP SL key can be enabled to run on a virtual machine.

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.

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.

Note:

The above license term options do not apply to Provisional Products. For additional information, see [Defining Provisional Products on page 155](#).

Defining Sentinel HASP Memory Data

When you define a Product in Sentinel HASP Business Studio, you can define the layout and contents of the memory data associated with the Product.

You can define areas (segments) in memory and enter data into them as required. You can also specify that data is entered in one or more of the memory segments at order time. You can select different colors for each segment to make it easy to identify them, and you can redefine the data and the layout as required.

You can select the memory type for each segment that you define, according to the type and purpose of the data you want to store:

- **Read/Write Memory:** Data that can be updated when the deployed, protected program 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 program is running but cannot be changed, such as the Product version number, text to be used in a “Welcome” message, fixed threshold values for counters, and so on.

Note:

You can use either or both types of memory to store and control licenses from your own licensing schemes.

The data defined in memory is written to the HASP memory of the Sentinel HASP protection keys together with the Features, license terms and other data defined for the Product.

Defining Provisional Products

You can define Provisional Products that can be distributed for use during a grace period or as trialware.

The properties of a Provisional Product are similar to those for a standard Product, with the following exceptions:

- **Locking Type:** Provisional Products do not require a locking type, since they can be activated and used for a limited period without a Sentinel HASP protection key.
- **License Terms:** Each Feature in a Provisional Product is automatically assigned a **Time Period** value of **30** days. This value can be changed to a value with the range of 1–90 days.

For additional information on the purpose and use of Provisional Products, see [Designating Products for Trial or Grace Period Use on page 137](#).

Provisional Products are not available for inclusion in customer orders. Users authorized to perform Development tasks can bundle Provisional Products for distribution. For additional information, see [Generating Bundles of Provisional Products on page 179](#).

Product Status Values

When a Product is first defined, its status is **Ready**. This status indicates that the Product is available to be included in orders. The Product details can be changed until it is included in an order, and if required, the Product can be deleted.

After a Product has been included in an order, its status changes to **In use**. This status indicates that the Product is available for use in additional orders, but the Product details cannot be changed and it cannot be deleted.

If a Product has been included in an order and is subsequently withdrawn from use, its status changes to **Obsolete**. An Obsolete Product can be restored for use.

Duplicating a Product

After you have defined a Product, you can easily define additional Products with similar details, using the **Duplicate Product** option in Sentinel HASP Business Studio. 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.

Note:

If you duplicate 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 not been included in any order, you can delete it. A Product cannot be deleted once it is included in at least one order. You can, however, withdraw the Product from use by marking it as **Obsolete**.

An Obsolete Product cannot be added to orders, but its details are maintained in Sentinel HASP Business Studio for tracking purposes, and it continues to be functional when already at the end user's site.

Restoring a Product

An Obsolete Product can be restored. 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 HASP Business Studio to cater for changing circumstances, such as the release of new software versions and changes in customer requirements.

Sentinel HASP Business Studio enables you to maintain your licensing plan by defining new Features and Products as required. In addition, you can use Sentinel HASP Business Studio 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 cater for changes in your company's software applications, in customer demand, in the market, and so on. 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 (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, and/or memory and license term updates

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 value for the license type by adding or subtracting days or 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 [on page 139](#)), 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 HASP protection keys that have already been deployed, it can be included in a *HASP Update* order. When the HASP Update is applied, the data for the Modification Product is added to the data for the original Product in the Sentinel HASP protection keys.

For additional information on defining and producing orders, see Chapter 10, *Sentinel HASP Orders, Production, and Development Tasks*.

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 acknowledgement 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 Chapter 10, *Sentinel HASP Orders, Production, and Development Tasks*.

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 (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 158.

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 HASP Business Studio.

A HASP 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 Chapter 10, *Sentinel HASP Orders, Production, and Development Tasks*.

Chapter 10

Sentinel HASP Orders, Production, and Development Tasks

The first part of this chapter is intended for users assigned the **Order Management** and **Production** roles in Sentinel HASP. It describes how to use Sentinel HASP Business Studio to manage and produce orders.

The final part of this chapter is intended for users assigned the **Development** role. It describes how to use Sentinel HASP Business Studio to perform development-related tasks, including generating bundles of Provisional Products and Sentinel HASP Run-time Environment installer files, and exporting definition files.

For an overview of Sentinel HASP Business Studio and for information on starting to use the application, see Chapter 7, *Introduction to Sentinel HASP Business Studio*.

In this chapter:

- Sentinel HASP Order Processing and Production
- Managing Orders
- Producing Orders
- Performing Development-related Tasks
- Enabling Trial Use and Grace Periods

Note:

This chapter provides high-level information on the order management, production, and development-related processes in Sentinel HASP Business Studio. For detailed practical instructions for using each function, see the Sentinel HASP Business Studio Help documentation.

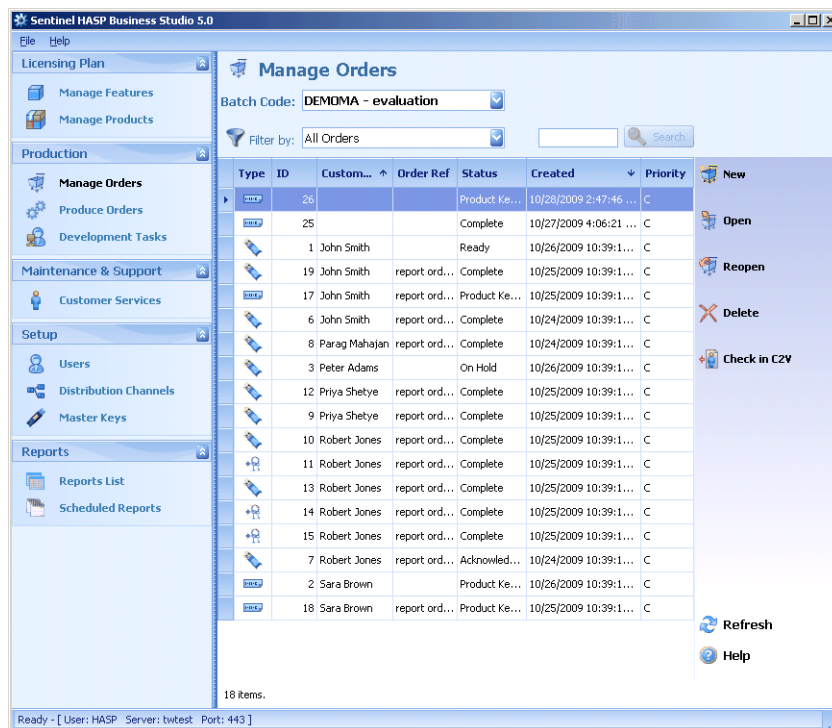
Sentinel HASP Order Processing and Production

An *order* is a request for Sentinel HASP items, and can be one of the following:

- A request for Products to be supplied with one or more Sentinel HASP protection keys
- A HASP Update that specifies changes to be made to the license terms and/or data stored in Sentinel HASP protection keys that have already been deployed

After Features and Products have been defined in Sentinel HASP Business Studio, orders can be processed and produced using the Production group of functions, including:

- Managing Orders
- Producing Orders
- Performing Development-related Tasks



The specific Sentinel HASP Business Studio functions you can access in the Production group of functions depend on the role assigned to you, as follows:

- If you have been assigned the **Order Management** role, you have access to both the Order Management and the Customer Services functions
- If you have been assigned the **Production** role, you have access only to the Order Production functions
- If you have been assigned the **Development** role, you have access only to the Development Tasks functions

Managing Orders

This section is intended for users assigned the **Order Management** role.

When you select the **Manage Orders** function in the Sentinel HASP Business Studio window, you can view the details of all customer orders associated with the selected Batch Code.

Note:

For additional information on Batch Codes, see [Personalized Vendor and Batch Codes on page 39](#).

You can perform the following tasks using the **Manage Orders** function:

- Define new customers
- Define orders
- Delete orders
- Process Customer-to-Vendor (C2V) information

Defining Orders

Before you start to define an order for a customer in Sentinel HASP Business Studio, ensure that you have the following information available:

- Details of the customer who placed the order (optional)
- The Products to be included in the order
- The required values for any license terms that have not yet been specified for the Products in the order
- The production requirements, according to the order type:
 - ◆ Order for HASP HL keys
 - ◆ Order for Product Keys
 - ◆ Order for HASP Update
- Additional order information (optional)

Note:

Sentinel HASP Business Studio generates a unique Order ID for each new order.

Defining the Customer for the Order

When you define the order in Sentinel HASP Business Studio, 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 edit the details of an existing customer. Sentinel HASP generates a unique Customer ID for each new customer.

Note:

You can also define a new customer using the Customer Services function.

Including Products in the Order

An order can contain one or more Products. All Sentinel HASP Products are associated with a Sentinel HASP Batch Code. You must specify the Batch Code in order to be able to select the Products to be included in the order.

Note:

Provisional Products (Products defined for use during a grace period or as trialware) are not available for inclusion in orders. The process of generating files containing Provisional Products is a Development task. For additional information, see [Generating Bundles of Provisional Products on page 179](#).

Each Product is assigned a *locking type when it is defined*. The locking type determines the level of Sentinel HASP protection and the type of Sentinel HASP protection key that can be supplied with the Product.

The locking type assigned to a Product may determine the type of order that can be produced:

- Products defined with the **HASP HL only** locking type can be included in orders for HASP HL keys, Product Keys, or for HASP Updates.
- Products defined with the **HASP SL only** locking type can be included only in orders for Product Keys or for HASP Updates.
- Products defined with the **HASP HL and HASP SL** locking type can be included in orders for HASP HL keys, Product Keys, or for HASP Updates

You cannot add a Product defined with the **HASP HL only** locking type and another Product defined with the **HASP SL only** locking type to the same order.

For additional information on locking types, see [Choosing the Protection Level for Your Products on page 134](#).

You can specify that Products that will be locked to HASP SL keys is to have clone detection enabled. For additional information, see [Protection Against Cloning on page 149](#).

Specifying License Term Values

When a Product is initially defined in Sentinel HASP Business Studio, the exact license term values for each Feature can be left unspecified. This enables you to include the same Product in different orders with different license term values.

In this case, the license values must be specified when each order 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 order:

- 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 order can be produced only after the license term values have been specified for all the Features in every Product included in the order.

Specifying Memory Data

When a Product is initially defined in Sentinel HASP Business Studio, memory data can be left unspecified. This enables you to customize memory data for each Product at order time. For example, customer-specific memory data can be added to the Product when an order is being processed.

Specifying an Order for HASP HL Keys

When an order for HASP HL keys is produced, the ordered Products are programmed (burned) on one or more HASP HL keys to be shipped to the customer. For additional information on HASP HL keys, see [HASP HL Keys on page 41](#).

When you define the order, you must specify the total number of HASP HL keys to be produced for the order.

Specifying an Order for Product Keys

A Product Key-based order enables you to produce activation strings for Sentinel HASP protection keys.

The ordered Products are associated with one or more Sentinel HASP Product Keys. A Product Key is a string of characters generated by Sentinel HASP Business Studio 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, the Sentinel HASP Business Studio Server verifies the Product Key and produces a Sentinel HASP protection key. The Sentinel HASP 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 a Product Key-based order, you must specify the following information:

- The number of Product Keys to be produced for the order
- 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 HASP SL keys, Product Keys can also optionally be used for activating software that is locked to HASP HL keys.

Note:

Before a HASP SL key can be used on an end user's computer, a Provisional Product must have been installed on the computer. When the Provisional Product is installed, it initializes the Sentinel HASP Run-time Environment, which is required for communication between the HASP SL key and the software. The process of generating files containing Provisional Products is a Development task. For additional information, see [Generating Bundles of Provisional Products on page 179](#).

Specifying a HASP Update Order

A HASP Update order specifies changes to be made to the license terms, Products, and/or data stored in Sentinel HASP protection keys that have already been deployed to end users. A HASP Update can be applied remotely to HASP HL keys or HASP SL keys, either using the Sentinel HASP Run-time API by calling the `hasp_update` function, or by using the Sentinel HASP Remote Update System utility.

When the HASP Update order is produced, a file containing the details of the changes is generated for each Sentinel HASP 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 HASP Remote Update System (RUS) utility

For additional information on RUS, see Chapter 12, *Sentinel HASP Remote Update System*.

When you define a HASP Update order, you must specify the total number of Sentinel HASP protection keys to be updated as a result of this order. You may also need to select the specific Sentinel HASP protection keys to be updated.

Locating the Sentinel HASP Protection Keys to Update

When you define a HASP Update order, you may need to select the specific Sentinel HASP protection keys to be updated. For example, the order may be for an organization with 100 Sentinel HASP protection keys, and this order is required to update the keys for only 10 specific users.

In Sentinel HASP Business Studio, you can:

- Display a list of the customer's Sentinel HASP protection keys
- View the contents of each key
- Select the keys to be updated

Note:

You cannot select more Sentinel HASP protection keys than the total number of keys specified in the **Order Details** area in the Production Order dialog box.

Optional Order Information

You can add the following optional information to the order:

- Order reference information that can identify the order in a different system, for example, an order number in your company's ERP system.
- A priority level, to indicate the urgency of the order for production purposes. The default is Priority **C**.
- A free-text comment that provides additional information about the order.

Adding the Order to the Production Queue

After you have specified all the necessary information for an order, you can produce it immediately or add it to the *production queue*. The *queue* is a list of all orders awaiting production.

Orders in the production queue can be selected for production according to the criteria determined by your organization.

Sentinel HASP Business Studio enables you to put on hold any orders that have not been completely defined, without losing the information that you may have already specified. You can open the order and continue to define the order details when convenient.

Order Status Values

When an order is first defined, its status is **On Hold**. This status indicates that the order is not yet in the production queue. The order details can be changed, and if required, the order can be deleted.

When an order is in the production queue, its status changes to **Ready**. This status indicates that the order is awaiting production. The details of a Ready order cannot be changed. However, it can be deleted.

In an order for Sentinel HASP Product Keys:

- After the Keys have been generated, the status of the order changes to **Product Keys generated**.
- If the order contains multiple Product Keys, as soon as at least one Product Key has been used to activate the protected software, the order status changes to **Produced**. The order contains additional Product Keys that have not yet been used.
- After all the Product Keys in the order have been used to activate the protected software, the order status changes to **Complete**.

In an order for updates or for HASP HL keys, after production of the order has been completed, the status of the order changes to **Complete**.

When verification that an order has been applied to a machine at the end user site has been received, the status changes to **Acknowledged**.

Processing C2V Information

C2V files contain protected information about the license terms and data stored in deployed Sentinel HASP protection keys. They do not contain private customer information.

C2V files can be generated using the Sentinel HASP Remote Update System (RUS) utility. For additional information on RUS, see Chapter 12, *Sentinel HASP Remote Update System*.

C2V information stored in HASP HL keys and in C2V files can be retrieved for use in connection with HASP Update orders.

When a C2V file or HASP 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 HASP Business Studio. The process of checking in the C2V information stores the data securely on the Sentinel HASP Business Studio Server, 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 HASP 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 HASP HL key, you can view similar information.

Note:

If you attempt to check in a C2V file for a HASP SL key, and the Sentinel HASP Business Studio Server 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 HASP SL keys, see *Protection Against Cloning on page 149*.

Formatting a HASP HL Key

You can format a HASP HL key to make it available for reuse. The process of formatting a HASP HL key deletes any orders that have been defined for the key but not yet produced. It also produces a V2C file that contains HASP Update information to be applied to the key using RUS. Applying the HASP 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 HASP HL keys
2. Order for Product Keys (HASP SL keys)
3. HASP Update order

Order Example 1: Order for HASP HL Keys

Scenario: A new customer, ABC Design, orders the **SafeNetCAD Office** Product from HQ Software with a license for 20 users.

Since the **SafeNetCAD Office** Product is defined with HASP HL key protection, the details for this order are defined as follows:

- **Customer:** ABC Design
- **Product:** SafeNetCAD Office
- **Order type:** HASP HL keys
- **Number of keys:** 20

When this order is produced, the **SafeNetCAD Office** Product license is programmed on 20 HASP HL keys, which are then shipped to the customer.

Order Example 2: Order for Product Keys (HASP SL Keys)

Scenario: On March 15, 2007, another customer, JL Optics, orders the **SafeNetCAD Home** Product, with a license for use on two computers.

The **SafeNetCAD Home** Product is defined with HASP 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:** SafeNetCAD Home
- **Expiration date for DRAW and SAVE:** March 15, 2008
- **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 **SafeNetCAD Home[Trial]** Provisional Product on the two computers before ordering the **SafeNetCAD Home** Product. As a result, the Sentinel HASP Run-time Environment for HASP 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 e-mail.

Two end users at JL Optics open the file and enter the Product Key as required on the HQ Software Website. The HQ Software customer interface application sends the Product Key to the Sentinel HASP Business Studio Server, which verifies the Product Key and returns a HASP SL key to the customer.

The HASP SL key is installed on the two computers at JL Optics with the license information, and the **SafeNetCAD Home** Product can be activated under the terms of the license.

Order Example 3: Order for HASP Update

Scenario: HQ Software informs ABC Design that a new version of **SafeNetCAD 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, **SafeNetCAD Office v.2.0**, ready for inclusion in customer orders.

Before defining the HASP Update order, HQ Software needs to receive C2V files for the five HASP HL keys to be updated. ABC Design uses RUS 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 HASP Update order for the Modification Product.

The details for this order are defined as follows:

- **Customer:** ABC Design
- **Product:** SafenetCAD Office v2.0
- **Order type:** HASP Update
- **Number of Sentinel HASP protection keys to be updated:** 5.

During the order definition process, the five HASP HL keys to be updated are selected from all the keys issued to ABC Design, according to the C2V files received.

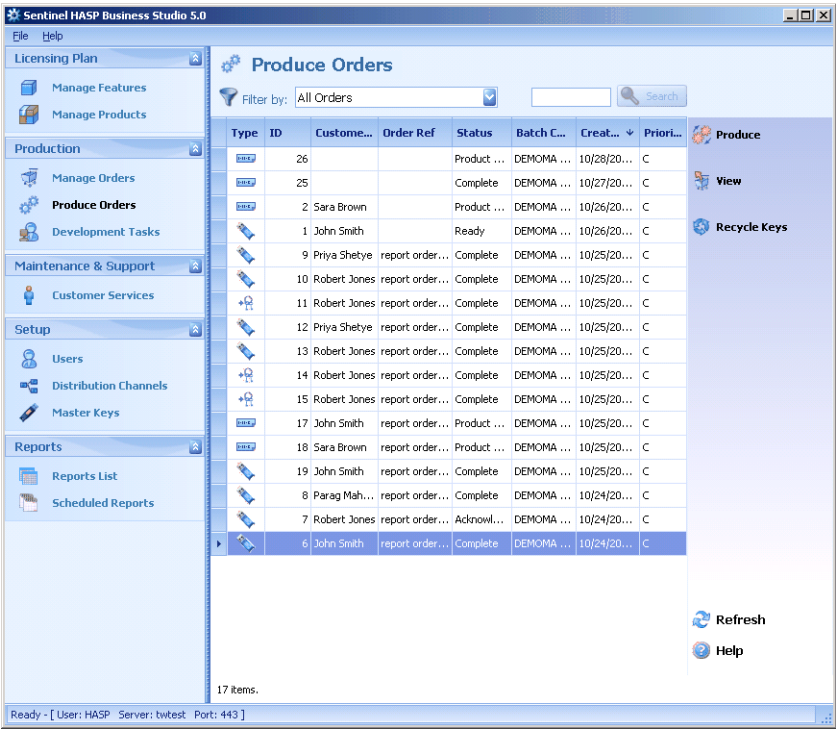
When this order is produced, a V2C file is generated for each selected HASP HL key and sent to the customer.

The selected five end users install the update on their HASP HL keys, using RUS. They are then able to activate the upgraded version of **SafeNetCAD Office** and to generate tailored reports.

Producing Orders

This section is intended for Sentinel HASP Business Studio users assigned the **Order Management** or **Production** role.

When you select the **Produce Orders** function in Sentinel HASP Business Studio, you can view the details of all orders awaiting production.



You can perform the following tasks using the **Produce Orders** function:

- Produce Orders
- View Orders

Note:

If you have been assigned both the **Order Management** and the **Production** roles, you can choose to produce an order immediately after you finish defining it.

The process of producing an order is determined by the type of order:

- Order for HASP HL keys
- Order for Product Keys
- Order for HASP Update

While producing any order, you can open the order and view its details.

Producing HASP HL Key Orders

Before you start to produce an order for HASP HL keys, Sentinel HASP Business Studio enables you to prepare the appropriate HASP HL keys to use for the order, by displaying:

- The HASP HL key types that are valid for the order
- The number of HASP HL keys to be produced, as specified in the order

Sentinel HASP Business Studio determines which HASP HL keys are valid for the order according to a number of factors, including:

- The license terms defined for the Features in the Products included in the order
- The data defined in memory for each Product
- The space required on the key to accommodate the order

For example, if the license terms for a Product in the order are based on a number of days or an expiry date, the order can be produced only on HASP HL keys with date and time monitoring capabilities, such as HASP HL Time.

Similarly, if the license terms for a Product in the order specify a number of concurrent instances in a network environment, the order can be produced only on HASP HL keys with network monitoring capabilities, such as HASP HL Net.

For additional information about HASP HL key types and their capabilities, see [Available Sentinel HASP Protection Keys on page 42](#).

Producing Orders for Product Keys

When you produce an order 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. For example, they can be sent to customers by Email, printed on the cover of a CD, and so on.

Producing HASP Update Orders

The order production process generates a file containing the HASP Update information for each Sentinel HASP protection key to be updated. After the files have been generated, they can be sent to the customer.

Before you generate the file, you must select the required location and the type of files to be generated for delivery to end users:

- *Vendor-to-Customer (V2C)* files that can be processed using the Sentinel HASP Remote Update System (RUS) utility
- Executable files (EXE) that contain V2C data and can be used as instructed by your company

For additional information on RUS, see Chapter 12, *Sentinel HASP Remote Update System*.

Note:

A default file location for V2C files may have been specified by the Sentinel HASP Administrator.

Performing Development-related Tasks

This section is intended for users assigned the **Development** role.

When you select the **Development Tasks** function in the Sentinel HASP Business Studio window, you can view a list of all development-related activities that have been performed for the selected Batch Code.

The most recent activities are displayed at the top of the list.

You can perform the following development-related activities in Sentinel HASP Business Studio:

- Generate bundles of Provisional Products
- Export data definitions to a file
- Customize the Sentinel HASP Remote Update System (RUS) utility
- Generate a customized Sentinel HASP Run-time Environment (RTE) installer file

Generating Bundles of Provisional Products

When a Product is defined in Sentinel HASP Business Studio, it can be specified as a Provisional Product for distribution as trialware or for use during a grace period.

Bundles of one or more Provisional Products can be shipped for use for a restricted period of time, (currently maximum 90 days).

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 HASP HL key or a HASP SL key.

For additional information on the purpose and use of Provisional Products, see [Designating Products for Trial or Grace Period Use on page 137](#).

The process of generating a bundle of Provisional Products involves:

- Selecting the Provisional Products to be included in the bundle
- Producing a file containing the Provisional Product license and Vendor library. This file can be:
 - ◆ An EXE file containing V2C data
 - ◆ A V2C file that can be used with the Sentinel HASP RUS utility. For additional information on RUS, see Chapter 12, *Sentinel HASP Remote Update System*.

The output file from this process must be installed on each end user's computer in order to:

- Create an initial Sentinel HASP Run-time Environment that enables your protected software to communicate with HASP SL keys.
- Enable a trialware or grace period license.

Note:

When a bundle of Provisional Products is installed on an end user's computer, a provisional HASP ID is generated. This is replaced by a Sentinel HASP protection key ID when a fully licensed Product is installed on that computer.

To simplify the installation process at end users' sites, it is recommended that you generate a Sentinel HASP Run-time Environment installer executable. You can embed the Run-time Environment installer in your software setup to create a ready-to-run Sentinel HASP protected and licensed application.

To generate a Sentinel HASP Run-time Environment installer executable, you need to specify the V2C file generated when a Provisional Product bundle is produced. An EXE file containing V2C data cannot be used to generate a Sentinel HASP Run-time Environment installer.

Generating the Sentinel HASP Run-time Environment Installer

You can generate a Sentinel HASP Run-time Environment installer that simplifies the installation process at end users' sites, for Provisional Products.

The input to this process is a V2C file that contains your vendor-specific data. For Provisional Products, the V2C file also contains the Provisional Product bundle data.

The output can be one of the following:

- An executable file that creates a Run-time Environment command-line installer
- A DLL that can be used with the Sentinel HASP Run-time Environment installer API
- A Mac PKG Sentinel HASP Run-time Environment installer

You can embed the Sentinel HASP Run-time Environment installer in your software setup to create a ready-to-run Sentinel HASP protected and licensed application.

Generating a Sentinel HASP Run-time Environment installer for Running a Product with a Detachable License

In order for a recipient machine to run an application using a detachable license, the Sentinel HASP Run-time Environment and vendor libraries must be installed. This condition is achieved by creating a Run-time Environment Installer that automatically installs these components. For additional information, refer to *Working with Detachable Licenses* in the *Sentinel HASP Business Studio Help* documentation.

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.

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 HASP Business Studio Help documentation.

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 HASP Business Studio.

Customizing and Branding RUS

RUS is a utility that can be distributed to end users to enable secure, remote updating of the license and memory data of Sentinel HASP protection keys after they have been deployed.

End users can invoke the RUS utility directly in order to generate a C2V file, or they can launch the utility by double-clicking an EXE file containing a license update.

Before you distribute RUS, you must customize it with the Batch Code associated with the Sentinel HASP 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 RUS is opened. For example, you may want to display your company name and information about your software.

The **RUS Branding** option in Sentinel HASP Business Studio enables you to associate the RUS utility with the selected Batch Code. You can also use the simple HTML editor provided to enter, format, and preview the text to be displayed in RUS.

It is recommended that you distribute your protected software with a customized and branded version of RUS.

For additional information on RUS, see Chapter 12, *Sentinel HASP Remote Update System*.

Enabling Trial Use and Grace Periods

This section provides examples that demonstrate the use of Provisional 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 a Provisional Product for Trial Use

Scenario: HQ Software decides to offer visitors to their Website 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 a Provisional Product. The license terms for the two Features were automatically set to **Time Period** with a value of **30** days.

The software developer at HQ Software defines a bundle of Provisional Products that contains the **HQ Design Demo** Product, and generates the bundle as a V2C file.

A Sentinel HASP Run-time Environment installer is then generated as an EXE file, using this V2C file as input.

The HQ Software Web master adds the EXE to the Website, with download 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 HASP HL key protection. The HASP 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 HASP HL keys arrive and are distributed to the end users.

For this purpose, a version of the **HQ Design Pro** Product is defined as a Provisional 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 automatically set to **Time Period** with a value of **30** days.

A bundle of Provisional Products is defined containing the **HQ Design Pro Grace** Product, and generated as a V2C file.

A Sentinel HASP Run-time Environment installer is then generated as an EXE file, using this V2C file as input.

The EXE file is sent to the customer, for distribution to the end users. End users can run the EXE, which installs the Sentinel HASP Run-time Environment and the **HQ Design Pro Grace** Product on their computers. They can then use the program for 30 days until they receive their HASP HL keys and can activate the full Product.

Sentinel HASP Administration and Customer Services

The first part of this chapter is intended for users authorized to perform Sentinel HASP Administration tasks. It describes how to use Sentinel HASP Business Studio to define user details, manage Sentinel HASP licenses and Sentinel HASP Master keys, and configure system settings.

The second part of this chapter is intended for users authorized to perform Sentinel HASP Customer Services tasks. It describes how to use Sentinel HASP Business Studio to view and edit customer details, and to perform manual Product activation for customers.

For an overview of Sentinel HASP Business Studio and for information on starting to use the application, see Chapter 7, *Introduction to Sentinel HASP Business Studio*.

In this chapter:

- Administration Tasks
- Customer Services

Note:

This chapter provides high-level information on the Administration and Customer Services processes in Sentinel HASP Business Studio. For detailed practical instructions for using each function in Sentinel HASP Business Studio, see the Sentinel HASP Business Studio Help documentation.

Administration Tasks

After you first install Sentinel HASP Business Studio in your organization, you can log in to Sentinel HASP using the default user name and password (**HASP**) provided for your use by SafeNet. By default, this user is authorized to perform all tasks in Sentinel HASP Business Studio, including Administration tasks.

Note:

The 'HASP' administrator details cannot be edited or deleted. Only the password can be changed.

After logging in to Sentinel HASP Business Studio the first time, it is recommended that you select the **Users** function and change your user password as soon as possible. If you want, you can change the roles assigned to you, but it is important that you retain the **Administration** role or assign this role to another user.

To be able to use Sentinel HASP with your company-specific Batch Codes and license, you must first introduce the Sentinel HASP Master keys provided for your use by SafeNet.

For additional information on Sentinel HASP Vendor keys, see [Personalized Vendor and Batch Codes on page 39](#).

For additional information on introducing Sentinel HASP Master keys, see [Maintaining Sentinel HASP Master Keys on page 188](#).

From time to time, you will need to renew your Sentinel HASP license, or to replenish your pool of activations and/or seats. You can schedule email notifications to be sent when it is time to renew or reorder, ensuring you uninterrupted use of Sentinel HASP.

For additional information about Sentinel HASP licenses, activations, and seats, see Appendix F, [Understanding the Sentinel HASP Master Key Licenses](#).

For additional information about configuring and scheduling email notifications, refer to the Sentinel HASP Business Studio Help documentation.

Note:

If you are evaluating Sentinel HASP Business Studio, you can use the provided **DEMOMA** Batch Code, which does not require a Sentinel HASP Master key.

You can now define additional Sentinel HASP users in your company, including assigning the users the appropriate roles and authorizing access to batches. For additional information, see [Maintaining User Details on page 187](#).

You can also view the system settings for Sentinel HASP Business Studio, and if required, change them to meet your company's requirements. For additional information, see [Configuring System Settings on page 190](#).

Note:

Non-administrator users can change these settings for their own use.

Maintaining User Details

When you select the **Users** function in Sentinel HASP Business Studio, you can view the Sentinel HASP details of all currently defined Sentinel HASP users.

You can perform the following tasks using the **Users** function in Sentinel HASP Business Studio:

- Define Sentinel HASP users
- Change user details and passwords
- Prevent user access

Defining Sentinel HASP Users

Before you start to define Sentinel HASP users, ensure that you have the following information available for each new user:

- The user name to be assigned to the user for the purpose of logging in to Sentinel HASP
- The full name of the user (optional)
- The password to be assigned to the user

Note:

Users can change their own passwords after logging in to Sentinel HASP Business Studio.

- The batches 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 *Sentinel HASP Business Studio Users and User Roles* on page 121.

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. However, if necessary, you can change the password for a user without knowing the current password. This is useful in the event that the user has lost or forgotten his/her password.

Preventing User Access

In certain circumstances, you may want to prevent a user from logging in to Sentinel HASP. If the user has left the company, for example, or will no longer be using Sentinel HASP, you can delete the user details.

If you want to prevent a user from accessing Sentinel HASP temporarily, without deleting their details, you can edit the details and specify that the user's access is *blocked*. The user will be unable to log in to Sentinel HASP until you change the user's details to *active*.

Maintaining Sentinel HASP Master Keys

When you select the **Master Keys** function in Sentinel HASP Business Studio, you can view the details of all available Sentinel HASP Master keys for the selected Batch Code that are currently connected to the Business Studio Server.

You can perform the following tasks using the **Batch Codes** function in Sentinel HASP Business Studio:

- Introduce Sentinel HASP Vendor keys
- Generate a C2V file for a selected Sentinel HASP Master key

- Apply a V2C file to a selected Sentinel HASP Master key, in order to:
 - ◆ update your Sentinel HASP license
 - ◆ replenish your pool of activations
 - ◆ replenish your pool of seats
- Edit the descriptive properties of Batch Codes
- Specify Mail Notification properties

Introducing Sentinel HASP Vendor Keys

The Sentinel HASP Master key(s) for your organization are introduced as part of the Sentinel HASP Business Studio Server installation process. You must have a separate Sentinel HASP Master key connected to each server on which Sentinel HASP Business Studio Server is installed.

You can introduce additional Sentinel HASP Vendor keys—Sentinel HASP Master keys or Sentinel HASP Developer keys—in order to enable Batch Codes for use with Sentinel HASP applications.

When you introduce a Sentinel HASP Vendor key, you can select the libraries for which you want to generate APIs. You may have the option to merge the APIs of multiple Batch Codes into a single library.

Generating a C2V File

When you submit an order for an update to your Sentinel HASP license, regardless of whether it is to renew the license or to replenish your pools of activations or seats, you need to generate a C2V file for the Sentinel HASP Master key that is to be updated. You then send the C2V to your Sentinel HASP supplier, together with your order. The C2V file contains encrypted information about the current status of your Sentinel HASP Master key, including its unique ID.

Editing Batch Code Properties

You can change the name and description of a selected batch as required. You cannot change the Batch Code itself.

Defining Mail Notification Properties

You can specify who is to receive notifications that your Sentinel HASP license and pools of activations or seats are about to expire. In addition, you can define the thresholds after which the notifications are sent.

Configuring System Settings

You can configure the following Sentinel HASP settings using the **System Settings** function in Sentinel HASP Business Studio. These settings affect all Sentinel HASP users:

- **Connection details:** By default, the details of the port and server used for the previous login to Sentinel HASP are retained and are reused for the next login. If other ports and/or servers were used to log in, they can be selected from a list. You can choose not to retain the connection details.
- **Connection history:** You can choose to delete all details of previous connections specified at login.
- **Default locking type:** You can select which **Locking Type** option will be displayed as the default when a Product is defined.
- **File locations:** You can specify the default location for files of the following types:
 - ◆ **C2V files:** These files contain customer-to-vendor information and are generated by the end user with the Sentinel HASP Remote Update System (RUS) utility. For additional information, see [Processing C2V Information on page 172](#).
 - ◆ **V2C files:** These files contain vendor-to-customer information related to orders for HASP Updates. For additional information, see [Specifying a HASP Update Order on page 170](#).
 - ◆ **Vendor Code files:** These files contain Vendor Code information. For additional information on Sentinel HASP Vendor keys and codes, see [Personalized Vendor and Batch Codes on page 39](#).
 - ◆ **Export files:** These files contain definition details exported for various purposes. For additional information, see [Exporting Definition Data on page 182](#).

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 HASP 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 returns a C2V file to confirm that the Product has been activated.

For additional information on C2V files, see [Processing C2V Information on page 172](#).

Sentinel HASP Remote Update System

This chapter describes the Sentinel HASP Remote Update System (RUS) utility and explains how to use RUS to update license data remotely for deployed Sentinel HASP protection keys.

Note:

You can also apply updates to deployed Sentinel HASP protection keys using the Sentinel HASP Run-time API, by calling the `hasp_update` function. For additional information, see the Sentinel HASP Run-time API Help documentation.

In this chapter:

- [RUS Overview](#)
- [RUS Workflow](#)
- [Using RUS](#)

RUS Overview

RUS is an advanced utility that enables secure, remote updating of the license and memory data of Sentinel HASP protection keys after they have been deployed. As part of the basic concept underlying Sentinel HASP, RUS facilitates ongoing licensing well after protection has been implemented. For additional information on Sentinel HASP concepts, see [Protect Once—Deliver Many on page 36](#).

RUS provides a simple and secure method of updating your licenses remotely, after you have delivered your protected software together with the Sentinel HASP protection keys. You simply need to update the license and deliver update files to your customers.

RUS enables you to receive information on the current status of Sentinel HASP licenses at your customers' sites, and to securely extend or reduce the functionality of these licenses, without recalling the Sentinel HASP protection keys.

Note:

All Sentinel HASP protection keys except the HASP HL Basic key can be updated using RUS.

RUS is an executable utility (`hasprus.exe`) that can be distributed to end users with your software.

It is important that you customize RUS with the Batch Code associated with the Sentinel HASP protection keys that you produce for your customers, before you distribute the executable to them. For additional information on Batch Codes, see [Personalized Vendor and Batch Codes on page 39](#).

You can use Sentinel HASP Business Studio to customize RUS with the required Batch Code, and also to brand the GUI to display your vendor-specific information to end users. For additional information, see [Customizing and Branding RUS on page 182](#).

RUS Workflow

When you deliver your Products to a customer, you can include a customized version of RUS with the installation package. You can also include the instructions for using RUS.

When a license update is required, you have the option of either retrieving customer licensing information from the Business Studio Server, or of requesting that a customer produces and sends you a Customer-to-Vendor (C2V) files for the Sentinel HASP protection keys to be updated. C2V files have a `.c2v` extension and contain information on the licensing and memory content of the Sentinel HASP protection keys.

When you receive C2V files from a customer, you check them in using Sentinel HASP Business Studio. For additional information, see [Processing C2V Information on page 172](#).

Regardless of whether you obtain the data from the Business Studio Server, 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 HASP Business Studio as HASP Update orders for delivery to the customer. For more information on defining HASP Update orders, see [Specifying a HASP Update Order on page 170](#).

The process of producing a HASP Update order generates a file for each Sentinel HASP 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 HASP Update order production process, see [Producing HASP Update Orders on page 178](#).

The output file is then delivered to the end user, who either runs the executable as instructed by you, or uses RUS 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 HASP HL keys to be updated.

ABC Design uses RUS to generate the C2V files and sends them to HQ Software. These files contain the current status of the license on the specific HASP HL keys.

HQ Software checks in the C2V files, defines a HASP 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 HASP Update on page 175](#).

The V2C files are sent by email to ABC Design. Each of the five end users applies the update to their HASP HL key using RUS, and returns a C2V file containing a confirmation receipt.

Using RUS

The RUS window consists of the following tabs:

- **Collect Key Status Information:** The parameters in this tab are used to collect information on the current status of the licenses in the Sentinel HASP protection key. The end user specifies a name and location for the generated C2V file. If more than one Sentinel HASP 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 HASP protection key.

Instructions for Customers Using Sentinel HASP RUS

The following sections contain information and instructions that you can customize and send to your customers.

Instructions for Using Sentinel HASP RUS

If you are using RUS with a HASP HL key, (hardware-based key) you must connect the key before performing either of the following procedures. RUS automatically locates any HASP SL keys (software-based keys) installed on your computer.

Collecting Sentinel HASP Protection Key License Data

You can use Sentinel HASP RUS to produce a Customer-to-Vendor (C2V) file containing information on the current status of the licenses in your Sentinel HASP protection keys. You can then send this file in order to receive a license update.

To retrieve the current license information from a Sentinel HASP protection key:

1. Launch Sentinel HASP RUS (`hasprus.exe`).
2. Click the **Collect Key Status Information** tab.
3. Click **Collect Information**. The Save key status as window is displayed.
4. Specify the directory where you want to store the C2V file. Enter a file name and click **Save**.

5. If more than one Sentinel HASP 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**.
6. The C2V file for the Sentinel HASP protection key is generated and saved in the required location. The file can now be sent for processing to produce an update.

Applying an Update

You can also use Sentinel HASP RUS to apply an update to the licenses stored in your Sentinel HASP protection keys.

To update the licenses in Sentinel HASP protection keys:

1. Launch RUS (`hasprus.exe`) or double-click the Vendor-to-Customer (V2C) 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.

2. Click the **Apply License Update** tab. (This might be the only tab displayed.)
3. If the **Update file** field is empty, browse to the directory where the update file (`.v2c` file) is located and select the file.
4. Click **Apply Update** to apply the new license data to the deployed Sentinel HASP protection key.

Generating Sentinel HASP Reports

This chapter describes the Reporting facility in the Sentinel HASP Business Studio.

In this chapter:

- Reports Facility Overview
- Permissions for Working With Reports
- Scheduling Reports
- Presentation Formats
- Export Formats
- Available Reports

Reports Facility Overview

The Sentinel HASP Reports facility provides you with the ability to produce reports with valuable business information, based on data in the Sentinel HASP database. With this tool, managers can 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 turn trial users into buyers.

The Sentinel HASP Reports facility connects directly to the Sentinel HASP database, and generates reports based on SQL queries.

The Sentinel HASP Reports facility 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.

Note:

The Sentinel HASP Reports facility is licensed separately from Sentinel HASP Business Studio. To obtain a license to use the Reports facility, contact your SafeNet representative.

The remainder of this chapter provides an overview of features and options available in the Reports facility.

For detailed information on operating the facility, refer to the online help in the Sentinel HASP Business Studio.

Permissions for Working With Reports

Access to the Reports facility is limited to Sentinel HASP Business Studio users who have been granted the role **Report Generation** or **Administrator**. Only these users can view reports directly in Sentinel HASP Business Studio. (The role **Report Generation** provides access only to the Reports facility, and only for the specific Batch Codes selected.)

However, using the scheduling option in the Reports facility, an authorized user can define a distribution list for each report. Each member of the distribution list receives the report by e-mail. The list can include Sentinel HASP Business Studio users (for whom an e-mail address has been specified in Sentinel HASP Business Studio) or any valid e-mail address.

No special authorization is required to receive reports by e-mail.

Scheduling Reports

An authorized Sentinel HASP Business Studio 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 e-mail each time it is generated.

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.

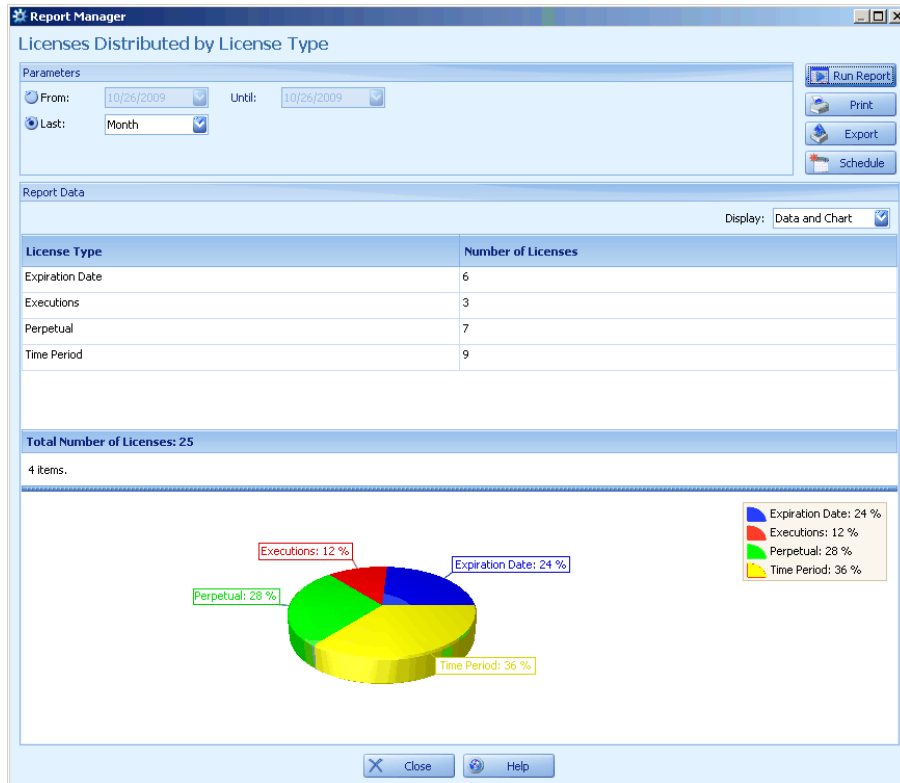
Note:

To generate scheduled reports using a non-Latin font (that is, using Japanese or Chinese), it is necessary to configure the Sentinel HASP Business Studio Server to use an appropriate TrueType font. For more information, see the *Sentinel HASP Installation Guide*.

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.

Tabular Display with Pie Chart



Export Formats

Each report can be exported from Sentinel HASP Business Studio or sent to the recipients in the distribution list in any of the following formats:

- Adobe Acrobat (PDF file)
- Microsoft Word (DOC file)
- Microsoft Excel (XLS file)
- Comma-separated values (CSV file)

Available Reports

The reports listed below are available in the Sentinel HASP Reports facility.

Report Name	Report Description
Activations	
Key Activations by Date	Lists all Product Key activations and summarizes them by date.
Customer	
Number of Keys by Customer and Key Type	Summarizes the number of HASP keys (by type) generated for each customer.
Order Type Distribution for a Specific Customer	Provides a breakdown of the different types of orders created for a specific customer.
Distribution Channel	
Number of Products by Distribution Channel	Summarizes the number of Products that are associated with each Distribution Channel.
Orders Produced by Distribution Channel	Lists all the orders created by users assigned to a specific Distribution Channel and provides a breakdown of the orders by order type.
Products Associated with a Specific Distribution Channel	Provides a list of all Products that are associated with a specific Distribution Channel.
License Key	
Keys with Expired Licenses	Displays a list of HASP keys with Features that have expired.
Keys with License About to Expire	Displays a list of HASP keys with Features that will expire in the near future.
Licenses Distributed by License Type	Summarizes the number of licenses produced by license type.

Report Name	Report Description
Master Key	
License Information by Master Key	Provides details regarding license information for a specific Master key.
Seats Consumed by Customer	Provides a summary by customer of the number of network seats consumed.
Order	
Customer Order Activities by Date	Lists all activities performed on orders and summarizes the activities by type and date.
Customer Orders by Product	Lists all orders and summarizes the number of orders by Product.
Customer Orders for a Specific HASP ID	Lists all orders associated with a specific HASP ID.
Product	
Number of Products by Product Type	Provides a summary of the number of Products for each Product type.
Production	
Number of HL Keys Burned by Date	Lists all HL keys burned and summarizes the number of keys burned by date.
Number of HL Keys Burned by Type	Lists all HL keys burned and summarizes the number of keys burned by HASP key type.

Part 4 Distributing Sentinel HASP Software

In this section:

- **Chapter 14: Distributing Sentinel HASP With Your Software**
Describes options for distributing required software to your end users.
- **Chapter 15: Sentinel HASP Admin Control Center**
Describes the configuration and management functionality of Sentinel HASP Admin Control Center, an end-user utility that enables centralized administration of HASP License Managers and Sentinel HASP protection keys.



Chapter 14

Distributing Sentinel HASP With Your Software

This chapter introduces options for distributing required software to your end users.

In this chapter:

- Sentinel HASP Software for End Users
- Distributing Sentinel HASP Run-time Environment

Sentinel HASP Software for End Users

Every Sentinel HASP 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.

Note:

Of all the software components described in this section, only the Sentinel HASP Run-time Environment *must* be present on the computer of each end user who will use protected software.

Protection-related Software

To enable Sentinel HASP protection keys to run and communicate with the protected application, the Sentinel HASP Run-time Environment must be installed on the computer of each end user who will use the application. There are a number of ways in which the Run-time Environment can be installed.

For more information, see *Distributing Sentinel HASP Run-time Environment* on page 209.

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.dylib libHASPJava.jnilib
	32-bit Linux	libHASPJava.so
	64-bit Linux	libHASPJava_x86_64.so
	All	Customized Run-time API dynamic libraries (copied automatically to the output directory by Sentinel HASP Envelope)

These native library files enable the protected application to communicate with the Sentinel HASP protection key.

Network Environment Management

Your end users can manage their network licenses online using Sentinel HASP Admin Control Center. Ensure that you send them the URL for accessing this application. For additional information, see Chapter 15, *Sentinel HASP Admin Control Center*.

Software for Updating Licenses

Sentinel HASP Remote Update System is distributed when remotely updating licenses in deployed Sentinel HASP protection keys. For additional information on this utility, see Chapter 12, *Sentinel HASP Remote Update System*.

Distributing Sentinel HASP Run-time Environment

Sentinel HASP Run-time Environment enables your protected software to run by communicating with Sentinel HASP protection keys.

The following sections describe the various options available for distributing the Sentinel HASP Run-time Environment to your end users.

- *Sentinel HASP Run-time Environment for Windows* on page 209
- *Sentinel HASP Run-time Environment for Mac* on page 214
- *Sentinel HASP Run-time Environment for Linux* on page 215

Sentinel HASP Run-time Environment for Windows

The following options are available for distributing Sentinel HASP Run-time Environment for Windows operating systems:

- Use Windows Update to download the Sentinel HASP Run-time Environment. An Internet connection is required for this process.

- Integrate installation of the Sentinel HASP Run-time Environment into your application's installer using either of the two options below:
 - ◆ Sentinel HASP Run-time Environment Merge module
 - ◆ Sentinel HASP Run-time Environment Installation API
- Deliver either of the following Sentinel HASP Run-time Environment installation utilities to your end users:
 - ◆ **HASPUserSetup.exe**: A GUI-based installer
 - ◆ **haspdinst.exe**: A command-line utility

Each of these methods is described in greater detail in this section.

Windows Update

If your end users are running the protected software on Windows XP or later platforms, and can access the Internet, they simply need to connect a Sentinel HASP protection key on their machines. The Sentinel HASP Run-time Environment is certified by Microsoft, and is therefore automatically downloaded from the Microsoft Update site.

When your end users connect a Sentinel HASP protection key:

1. The system informs them that a new component has been detected.
2. The Sentinel HASP Run-time Environment is automatically installed.
3. The LED on the Sentinel HASP protection key lights up, indicating that the installation process is complete.

Sentinel HASP Run-time Environment Merge Module

The Sentinel HASP 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 HASP 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 HASP Run-time Environment.

The benefits of using the Sentinel HASP 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 HASP 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 page 212](#).

Implementation Requirements

Before including the Sentinel HASP merge module in your installer, review the following requirements:

- The Sentinel HASP 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 HASP Run-time 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 our development environment.
- Before compiling the MSI project, change the path to external files to match your development environment

Implementation

Implementation of Sentinel HASP 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 HASP installation merge module.

The `haspds.msm` merge module can be found in:
...\\Program Files\\SafeNet\\Sentinel HASP\\API\\Runtime\\Install\\MSI

Note:

- Do not alter the versioning data in the default merge module, or 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 case the `haspds_windows.dll` will be executed.

Sample Merge Module Installer

A sample MSI installer containing the Sentinel HASP 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 HASP 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 requisite administrator rights to install the Sentinel HASP Run-time Environment
- Stops a running HASP License Manager service before the Run-time Environment is installed, and re-starts the service after the installation is complete.
- Installs or removes Sentinel HASP Run-time Environment

The sample installer can be found in:

...\Program Files\SafeNet\Sentinel HASP\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 HASP Run-time Environment Installer API

Use the Sentinel HASP Run-time Environment installer API to integrate the installation process into your custom setup application. For additional information, see the separate help file in the ...\\Program Files\\SafeNet\\Sentinel HASP\\API\\RuntimeInstall\\ directory.

haspdinst.exe

haspdinst.exe is a command-line utility that installs the Sentinel HASP Run-time Environment. Following installation of Sentinel HASP Vendor Suite, the file is located in ...\\Program Files\\SafeNet\\Sentinel HASP\\Redistribute\\Runtime Environment\\cmd Install. You can distribute this standalone application to your end users.

To install the Sentinel HASP Run-time Environment using haspdinst.exe:

- At the command-line prompt, type `haspdinst -i`.

HASPUserSetup.exe

HASPUserSetup.exe is a GUI-based installation program to independently install the Sentinel HASP Run-time Environment. Following installation of Sentinel HASP Vendor Suite, the file is located in ...\\Program Files\\SafeNet\\Sentinel HASP\\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.

Sentinel HASP Run-time Environment for Mac

Distribute the Sentinel HASP daemons—**aksusbd** and **hasplmd**—to end users running protected and licensed applications on Mac OS X platforms.

All the Sentinel HASP software for Mac that is required for distribution to end users is provided in the MacOS/Redistribute/ directory on your Sentinel HASP installation DVD.

Options for Distributing the Sentinel HASP Daemons

The software required to distribute the daemons is provided in the MacOS/Redistribute/ directory on the Sentinel HASP installation DVD.

Multiple options are available for distributing the Mac daemons to end users. The following two options are described:

- Installer Distribution Using a Multi-packager
- Installer Scripts

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 **HASP SRM RTE Installer.pkg** in the mpkg.

To locate the HASP SRM RTE Installer.pkg:

1. In the MacOS/Redistribute/ directory, double-click **HASP SRM RTE Installer.dmg**. The file opens.
2. Click the **HASP SRM Runtime Environment** icon and select **Show Original**. The .Packages window opens and **HASP SRM RTE Installer.pkg** is displayed.

For additional information, see the **welcome.rtf** file provided in the MacOS/Redistribute/ directory.

Installer Scripts

Installation scripts are provided in MacOS/Redistribute/ on the Sentinel HASP installation DVD. Open the directory and click **HASP SRM RTE Installer Scripts.dmg**. A new volume named **HASP SRM RTE 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 **HASP SRM RTE Installer Scripts** volume.

Sentinel HASP Run-time Environment for Linux

Distribute the Sentinel HASP daemons—**aksusbd** and **hasplmd**—to end users running protected and licensed applications on Linux platforms. Without the daemons, the end user's system will not recognize the connected HASP keys, and the protected applications will not run.

All the Sentinel HASP software for Linux that is required for distribution to end users is provided in the Linux/Redistribute/ directory on your Sentinel HASP installation DVD.

Using the Installer Scripts to Distribute the Sentinel HASP Daemons

Open the Linux/Redistribute/Runtime/script directory. The directory contains **dinst** (install) and **dunst** (uninstall) scripts and the Sentinel HASP Run-time Environment.

You can integrate the scripts in your installer. The scripts are not configurable.

Using the RPM to Distribute the Sentinel HASP Daemons

This option is available for SuSE and RedHat Linux.

Open the Linux/Redistribute/Runtime/RPM directory. The directory contains the Sentinel HASP Run-time Environment and the following files:

- Use `aksusbd-redhat-1.14-3.i386.rpm` for RedHat
- Use `aksusbd-suse-1.14-3.i386.rpm` for SuSE

Chapter 15

Sentinel HASP Admin Control Center

Sentinel HASP Admin Control Center is a customizable, Web-based, end-user utility that enables centralized administration of HASP License Managers and Sentinel HASP protection keys. This chapter describes the configuration and management functionality.

In this chapter:

- Introduction to Admin Control Center
- Launching Admin Control Center
- Admin Control Center Interface
- Administrator's Workflow
- Applying Basic Configuration Changes Globally
- Customizing Admin Control Center Look and Feel

Introduction to Admin Control Center

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. It has been engineered in a way that provides both flexibility and customizability, making it a useful add-on to your Sentinel HASP-protected and licensed software.

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 integrate seamlessly into other applications, corporate styles, and so on. 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.

Launching Admin Control Center

Admin Control Center is installed as part of the Sentinel HASP 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 HASP License Manager residing on your own machine, type **http://localhost:1947**.

Admin Control Center Interface

When you launch Sentinel HASP Admin Control Center, the Web interface displays a number of Administration Options on the left of the page. The context-sensitive Help document, which forms part of Sentinel HASP Admin Control Center, provides information about the fields for each option. Note that the options relate to the License Manager on the machine whose name or IP address is in the title bar of Admin Control Center. The following options are available:

- **HASP Keys** enables you to identify which Sentinel HASP 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 HASP 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.
- **Features** enables you to view a list of the Features that are licensed in each of the Sentinel HASP 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 HASP 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 HASP protection key in the field and, in the case of HASP 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 HASP License Manager is running.

- **Configuration** enables you to specify certain operating settings for Sentinel HASP Admin Control Center running on the connected machine. You can set parameters relating to user access, access to remote HASP 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 HASP 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 HASP License Manager is running.
- **Help** displays the Help documentation for Sentinel HASP Admin Control Center. Context-sensitive Help is available within each of the functions described above, by clicking the **Help** link at the bottom of the page.
- **About** provides information about the version of HASP License Manager, and a link to the SafeNet, Inc.' Website.
- **Country Flags** enables you to change the language of the user interface by clicking on the flag of the country appropriate to the language you require. Languages other than English can be downloaded from the Sentinel website.

Administrator's Workflow

When you first launch Admin Control Center, the utility is pre-configured 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 on this 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 HASP License Managers
- Access privileges from remote client machines to a HASP License Manager on the current machine

Configuration Considerations

Before you make certain configuration changes to Admin Control Center, it is recommended that you consider their implications. This section provides a guide to assist you in this process.

Basic Settings

Should you choose to have Admin Control Center create an access log, consider the scope of the information that you want recorded in the log.

It is possible to customize the log template to define what information you want the log to record.

Customizing Log Parameters

You can specify the data that is displayed in a log file by editing the log parameters.

Access the Edit Log Parameters page by clicking **Edit Log Parameters** in the **Basic Settings** tab of the Configuration page.

The Edit Log Parameters page contains two fields. The upper field displays the parameters of the current template. The **Available tags for log** field lists all the tags and their descriptions. By selecting a tag and clicking **Add**, the tag is appended to the text in the upper field.

The default log parameters are used the first time that Admin Control Center is launched and will also be used if an empty template is submitted. The default template is:

```
{timestamp} {clientaddr}:{clientport} {clientid} {method}
{url} {function} ({functionparams} result
({statusCode}){newline}
```

You can reset your log parameters to the factory default by clicking **Set Defaults** in the Edit Log Parameters page. The default log parameters will also be used if there is no data in the upper field of the Edit Log Template page.

Managing Access to HASP License Managers

Managing Access to HASP License Managers is performed in the **Users** and **Access from Remote Clients** tabs in the Configuration page. The following paragraphs discuss the issues that you need to consider when setting these parameters.

Users

The user restrictions that you define are evaluated in the order in which they are specified, and the evaluation process stops when the first match is found. You therefore need to take care that the restrictions are listed in an order that satisfies this logic.

The value `allow=all@all` is implicitly added to the end of the list. According to the logic just described, if this value was at the beginning of the list, all subsequent restriction values would be ignored.

Additional information about defining restriction values is provided in the relevant Help page in Admin Control Center.

Access from Remote Clients

When you define criteria relating to the remote machines that can access HASP License Manager on the current machine, you need to define access restrictions. The remote client access restrictions that you define are evaluated in the order in which they are specified, and the evaluation process stops when the first match is found. You therefore need to take care that the restrictions are listed in an order that satisfies this logic.

The value `allow=all` is implicitly added to the end of the list. According to the logic just described, if this value was at the beginning of the list, all subsequent restriction values would be ignored.

Additional information about defining remote client access restriction values is provided in the relevant Help page in Admin Control Center.

Searching for License Managers

Managing the locations to search for HASP License Managers is performed in the **Access to Remote License Manager** tab in the Configuration page. The following paragraph discusses the issues that you need to consider when setting these parameters.

Access to Remote License Managers

When you define criteria relating to the machines that may be searched for HASP 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. In this case, it is necessary to specify each machine that may be searched—by specifying either its name or its IP address.

Configuring Detachable License Definitions

In Sentinel HASP Business Studio, it is possible to flag network-based licenses for Features in Products that will be locked to HASP SL keys as being *detachable*. This means that the Product has the ability to be temporarily detached from a pool of network seats and to be 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.

In the **Detachable License** tab of the Configuration page, you can enable this functionality and specify criteria relating to the number of licenses that may be detached from the pool of network seats, together with the maximum period for which the licenses may be detached. 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.

Configuring Log File Output

You can specify the information that you want to be displayed in your log files from the Basic Settings tab in the Configuration page. When you click the **Edit Log Parameters** button, the Edit Log Parameters tab opens and all tags that can be used to configure the output logs are displayed, together with their descriptions.

The default log parameters are used the first time that Admin Control Center is launched and will also be used if an empty template is submitted. The default template is:

```
{timestamp} {clientaddr}:{clientport} {clientid} {method}  
{url} {function} ({functionparams} result  
({statuscode}){newline}
```

Diagnostics

The Diagnostics page enables you to view and extract operating information for the HASP 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 relevant Help page in Admin Control Center.

Applying Basic Configuration Changes Globally

You can make configuration changes to Admin Control Center, and deploy them to multiple machines on the network, as follows:

1. Make the required changes on one machine, using the configuration page of Admin Control Center. The changes are saved in the `hasplm.ini` file.
2. Locate the `hasplm.ini` file using one of the following methods:
 - ♦ Read the full path name of `hasplm.ini`, which is located at the bottom of the Configuration page. The HASP base directory is the directory in which the `hasplm.ini` file resides.
 - ♦ 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).
3. Copy `hasplm.ini` and use it to overwrite `hasplm.ini` on all the other machines on the network.

Customizing Admin Control Center Look and Feel

You can change the language, displayed information, appearance, and behavior of Admin Control Center so that it will integrate into other applications, your organization's corporate styles, and so on.

The Admin Control Center GUI 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 HASP base directory.

To create a directory for a custom template:

1. Locate the **templates** directory inside the HASP base directory. The HASP base directory is located in:

...\\Program Files\\Common Files\\Aladdin Shared\\HASP (Windows)

/var/hasplm (Mac)

/etc/hasplm (Linux)

To determine the precise location of this directory on your system:

- ◆ Open Admin Control Center on your local machine (<http://127.0.0.1:1947>). Under Administration Options, click **Configuration**. Read the full path name of `hasplm.ini`, which is located at the bottom of the page. The HASP base directory is the directory in which the `hasplm.ini` file resides.
 - ◆ If the folder is empty, click **Submit**. The base path of `hasplm.ini` is updated at the bottom of the Configuration page.
2. Add **\\Aladdin Shared\\HASP\\templates\\<your_template_folder_name>** to the directory. For example, using an English version of Windows XP, the full path is
C:\\Program Files\\Common Files\\
Aladdin Shared\\HASP\\templates\\<your_template_folder_name>
-
- Note:**
- You can create multiple templates inside your **templates** directory.
 - Each time HASP 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 unrequired files.
-
3. Restart the HASP License Manager,
OR
Call http://127.0.0.1:1947/action.html?reload_templates to reload the new template.
 4. To verify your customized template, from a browser on your local machine, open http://127.0.0.1:1947/<your_template_folder_name>.

Writing Templates

A template is an ASCII text file (may be HTML, but also XML, CSV, and so on) that contains place holders (*tags*) for variables that are inserted by the HASP 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 HASP protection keys, Features, sessions, and so on). 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

...\Program Files\SafeNet\Sentinel HASP\Docs\Manuals & Tutorials\
Admin Control Center Customization\templates.

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 HASP 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—{this}.

For additional assistance, refer to the sample templates in

...\Program Files\SafeNet\Sentinel HASP\Docs\Manuals & Tutorials\
Admin Control Center Customization\templates.

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 HASP 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 HASP Admin Control Center functionality into your own Web application, possibly with the use of (i)frames, and so on.

Sample CSV Output

This section provides a sample CSV output. Such output is useful for importing the data into spreadsheets, databases, and so on.

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`. `_int_` denotes 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 the 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
[server name]:1947 Provides a shortcut to the main Admin Control Center page	[server name]:1947/_int_/index.html
[server name]:1947/corporate.html Automatically switches to the internal template. (_ini_) is set when no template has been specified	[server name]:1947/_int_/corporate.html
[server name]:1947/csv/devices.txt Doesn't change because the template (csv) and file name are specified	[server name]:1947/csv/devices.txt
[server name]:1947/sample 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 HASP Admin Control Center — it automatically redirects to the index page.

Part 5 Sentinel HASP Licensing Models

In this section:

- **Chapter 16: Sentinel HASP Licensing Models: Overview**
Provides an overview of Sentinel HASP Licensing models.
- **Chapter 17: Sentinel HASP Licensing Models: Description of Models**
Provides a detailed description of the various Sentinel HASP Licensing models that you can use to distribute your software.



Sentinel HASP Licensing Models: Overview

In this chapter:

- Introduction
- Sentinel HASP Licensing

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 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 HASP 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 HASP'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 models.

Sentinel HASP Licensing

Sentinel HASP offers a wide range of options and unprecedented flexibility for making and revising both licensing and protection strategies. Virtually any licensing model can be created—supported by the following fundamental Sentinel HASP concepts, technologies and applications:

- *Protect Once—Deliver Many™*

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 HASP, the software vendor can choose the device to which the protected software and license are locked—either to one of the many hardware-based HASP HL keys, or to a specific computer by means of a versatile software-based HASP SL key. The required level of protection, the licensing model, and the manner in which the software will be accessed and used collectively determine the most appropriate type of Sentinel HASP protection key. Locking the license to a hardware-based HASP HL key provides the strongest security.

- *Sentinel HASP Remote Update System (RUS)*

RUS provides a simple and secure method of remotely updating the licenses on deployed Sentinel HASP protection keys. Using RUS, software vendors can renew, extend, revise or revoke a license.
- *LicenseOnChip® and UpdateOnChip*

When a license is supplied on a hardware-locked HASP HL key, the licensing logic is embedded in the key's chip, employing Sentinel HASP'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 HASP Business Studio 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 HASP Envelope, the Sentinel HASP Run-time API, or a combination of both. The considerations for choosing a protection method are provided in [Determining the Best Protection and Licensing Method](#) on page 238.
- *Detachable Licenses*

A detachable license is available for Products that are locked to HASP 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 HASP offers two software protection methods that establish an inherent link between the protected software, the license, and the intelligence contained in a specific Sentinel HASP protection key.

- **Envelope-based protection (automatic)**
Sentinel HASP Envelope automatically wraps software in a protective shield and validates the licensing terms. Sentinel HASP 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 HASP Run-time 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 HASP functions and calls so that licensing terms are validated automatically, or can apply a customized license validation mechanism in order to implement specialized licensing models.

Most licensing 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 HASP API. Each licensing 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 envelope your application. 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 Chapter 8, *Preparing Your Sentinel HASP Licensing Plan*.

About This Section

This section describes a wide variety of licensing models, and provides guidelines for implementing them using Sentinel HASP. The licensing models include:

- Evaluation licenses (trialware or demoware)
- Component-based licenses
- Metered licenses
- Locked licenses
- Mobile licenses
- Network licenses
- Sales-assisting licenses
- Perpetual licensing

Note:

This section provides an outline of how to use Sentinel HASP to implement the described licensing models. For detailed instructions on how to protect and license your software, refer to earlier sections in this book and to the integral help system included in each of the Sentinel HASP applications.

How to Use the Licensing Models

Each licensing model in this section is introduced with a legend that describes the following:

- **Sentinel HASP functionality**—the Sentinel HASP functionality that enables creation of the described licensing model
- **Software distribution method**—the available methods for software distribution when using the described licensing model (physical package or electronic distribution)
- **Applicable key types**—the Sentinel HASP protection keys that can be used to implement the licensing model
- **Protection method**—the Sentinel HASP protection methods (Sentinel HASP Envelope or the Sentinel HASP Run-time API) that can be used to implement the licensing model

For example:

Sentinel HASP Functionality	Manages the maximum number of software executions
Software Distribution Method	<ul style="list-style-type: none">■ Physical package■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none">■ All HASP HL keys except HASP HL Basic■ HASP SL
Protection Method	<ul style="list-style-type: none">■ Envelope-based automatic implementation■ API-based automatic implementation

The legend is followed by:

- A short description of the licensing model
- Guidelines for implementation using Sentinel HASP

Sentinel HASP Licensing Models: Description of Models

This chapter provides a detailed description of the many types of licensing models that you can define using Sentinel HASP.

In this chapter:

- Evaluation Licensing Models
- Component-based Licensing Models
- Metered Licensing Models
- Locked Licenses
- Mobile Licenses
- Network Licensing Models
- Sales Boosting Licensing Models
- Perpetual Licensing model

Evaluation Licensing Models

Evaluation licensing 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 HASP Business Studio to create and produce the new license. The evaluation license is then seamlessly converted to a purchased license at the end-user site, using Sentinel HASP RUS.

The evaluation licensing models described below are:

- *Trialware*
- *High-security Time-limited Evaluation*
- *Demoware*

Trialware

Sentinel HASP Functionality	Creates a time-limited, software-based trialware license
Software Distribution Method	<ul style="list-style-type: none">■ Physical package■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none">■ HASP SL
Protection Method	<ul style="list-style-type: none">■ Envelope-based automatic implementation■ API-based automatic implementation

Description

Trialware is fully functional software that is made available for a limited time period (currently for any period between 1 and 90 days) 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 website, and on media such as a CD.

The time-limited trialware license does not use a dedicated Sentinel HASP 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

- Select the executable file that you want to license, and determine by which Feature ID it will be identified.
- Select your protection method:
 - ◆ *Envelope-based automatic implementation*
Envelope the executable file using Sentinel HASP Envelope, specifying its Feature ID.
 - ◆ *API-based automatic implementation*
In your code, insert a Sentinel HASP Run-time API Login call to the Feature ID.
- Create a Provisional Product in Sentinel HASP Business Studio, including the Feature IDs you defined.
- Distribute your trialware with Sentinel HASP Run-time Environment.
- When a fully licensed product is purchased, provide the end user with the appropriate Sentinel HASP protection key programmed with the license.

High-security Time-limited Evaluation

Sentinel HASP Functionality	Manages the period over which your software can be activated
Software Distribution Method	<ul style="list-style-type: none"> ■ Physical package
Applicable Key Types	<ul style="list-style-type: none"> ■ HASP HL Time ■ HASP HL NetTime
Protection Method	<ul style="list-style-type: none"> ■ Envelope-based automatic implementation ■ API-based automatic implementation

Description

The time-limited evaluation software is distributed, protected with a HASP HL key for maximum security. Due to the extra cost of providing software with a hardware-based HASP HL key, this evaluation method is suitable for high-end software or for software with a high evaluation-to-purchase conversion rate.

Implementation

- Select the executable file that you want to license, and determine by which Feature ID it will be identified.
- Select your protection method:
 - ◆ *Envelope-based automatic implementation*
Envelope the executable file using Sentinel HASP Envelope, specifying its Feature ID.
 - ◆ *API-based automatic implementation*
In your code, insert a Sentinel HASP Run-time API Login call to the Feature ID.
- Create the evaluation Product in Sentinel HASP Business Studio and define the expiration date for each Feature ID included in the Product.
- Distribute the evaluation software with a HASP HL key programmed with the license.
- Create the licensed Product in Sentinel HASP Business Studio and define the required licensing terms for each Feature ID included in the Product.
- When a fully licensed product is purchased, update the HASP HL key using Sentinel HASP RUS.

Execution-limited Evaluation

Sentinel HASP Functionality	Manages the maximum number of software executions
Software Distribution Method	<ul style="list-style-type: none">■ Physical package■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none">■ All HASP HL keys except HASP HL Basic■ HASP SL
Protection Method	<ul style="list-style-type: none">■ Envelope-based automatic implementation■ API-based automatic implementation

Description

Evaluation software that is restricted to a pre-determined number of executions. The evaluation software can be distributed with a HASP SL key—for example, via a website or on a demo CD. Alternatively, it can be distributed with a HASP HL key, providing maximum security.

Note:

- Using a HASP 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 HASP 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 HASP HL Time or HASP HL NetTime key must be used.
-

Implementation

- Select the executable file that you want to license, and determine by which Feature ID it will be identified.
- Select your protection method:
 - ◆ *Envelope-based automatic implementation*
Envelope the executable file using Sentinel HASP Envelope, specifying its Feature ID.
 - ◆ *API-based automatic implementation*
In your code, insert a Sentinel HASP Run-time API Login call to the Feature ID.
- Create the evaluation Product in Sentinel HASP Business Studio and define the permitted number of executions for each Feature ID included in the Product.
- Distribute the evaluation software with a Sentinel HASP protection key programmed with the license.
- Create the licensed Product in Sentinel HASP Business Studio, defining the licensing terms for each Feature ID included in the Product.
- When the end user purchases a fully licensed product, update the Sentinel HASP protection key using Sentinel HASP RUS.

Demoware

Sentinel HASP Functionality	Manages active and inactive software functionality
Software Distribution Method	<ul style="list-style-type: none">■ Physical package■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none">■ All HASP HL keys except HASP HL Basic■ HASP SL
Protection Method	<ul style="list-style-type: none">■ 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 HASP SL key (for example via a website or on a demo CD), or with the superior protection of a HASP 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 HASP 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 HASP HL Time or HASP HL NetTime key must be used.

Implementation

- Select the software functions that you want to license separately, and determine by which Feature ID they will be identified.
- In your code, insert a Sentinel HASP Run-time API Login call to the Feature ID.
- Create two Products in Sentinel HASP Business Studio:
 - ◆ 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.
- Envelope your software for additional security (optional).
- Distribute the demoware.
- When the end user purchases the software, send a Sentinel HASP protection key programmed with the full license.

Component-based Licensing 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 HASP, 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 licensing models described below are:

- *Module-based (Suites)*
- *Feature-based*
- *Time-limited Rental*
- *Phased Rental*
- *Micro-rental*
- *Subscription*
- *Pay-by-Peak Time (Peak Time)*
- *Time-based Overdraft*
- *Standard Counter*
- *Phased Counter*
- *Capacity (CPU/Memory/Disk)*

Module-based (Suites)

Sentinel HASP Functionality	Manages licensing of individual executables
Software Distribution Method	<ul style="list-style-type: none"> ■ Physical package ■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none"> ■ All HASP HL keys except HASP HL Basic ■ HASP SL
Protection Method	<ul style="list-style-type: none"> ■ Envelope-based automatic implementation ■ API-based automatic implementation

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 HASP protection key.

Implementation

- Select the executable files that you want to license separately, and determine by which Feature ID they will be identified.
- Select your protection method:
 - ◆ *Envelope-based automatic implementation*
Envelope the executable file using Sentinel HASP Envelope, specifying its Feature ID.
 - ◆ *API-based automatic implementation*
In your code, insert a Sentinel HASP Run-time API Login call to the Feature ID.
- In Sentinel HASP Business Studio:
 1. Create one or more Products.
 2. Include the required Feature IDs in each Product.
 3. Define the appropriate license terms for each Feature—for example, the number of executions, expiration date or concurrency.
- Distribute your software suite with the appropriate Sentinel HASP protection key programmed with the license.

Feature-based

Sentinel HASP Functionality	Manages licensing of separate functional components
Software Distribution Method	<ul style="list-style-type: none"> ■ Physical package ■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none"> ■ All HASP HL keys except HASP HL Basic ■ HASP SL
Protection Method	<ul style="list-style-type: none"> ■ 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

- Select the software functions that you want to license separately, and determine by which Feature ID they will be identified.
- In your code, insert a Sentinel HASP Run-time API Login call to each Feature ID.
- In Sentinel HASP Business Studio:
 1. Create one or more Products.
 2. Include the required Feature IDs in each Product.
 3. Define the appropriate license terms for each Feature—for example, number of executions, expiration date or concurrency.
- Envelope your software for additional security (optional).
- Distribute your software with the appropriate Sentinel HASP protection key programmed with the license.

Metered Licensing Models

In recent years, licensing 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 license 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, counter-based overdraft.

Time-limited Rental

Sentinel HASP Functionality	Manages the time period over which your software can be used
Software Distribution Method	<ul style="list-style-type: none"> ■ Physical package ■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none"> ■ HASP HL Time ■ HASP HL NetTime ■ HASP SL
Protection Method	<ul style="list-style-type: none"> ■ Envelope-based automatic implementation ■ API-based automatic implementation

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 pre-determined expiration date.

End users can monitor the remaining time using Sentinel HASP Admin Control Center, and can order a license renewal before the license expires. License renewal is implemented using Sentinel HASP RUS.

Note:

You can also specify a licensing period that is shorter than one day, as described in [“Micro-rental” on page 253](#)

Implementation

- Select the executable file that you want to license, and determine by which Feature ID it will be identified.
- Select your protection method:
 - ◆ *Envelope-based automatic implementation*
Envelope the executable file using Sentinel HASP Envelope, specifying its Feature ID.
 - ◆ *API-based automatic implementation*
In your code, insert a Sentinel HASP Run-time API Login call to the Feature ID.

- In Sentinel HASP Business Studio, create a Product that includes the Feature ID and define either an expiration date or the number of days until expiration.
- Distribute your software with the appropriate Sentinel HASP protection key programmed with the license.
- Renew the license remotely using Sentinel HASP RUS.

Phased Rental

Sentinel HASP Functionality	Manages the time period over which your software can be used
Software Distribution Method	<ul style="list-style-type: none">■ Physical package■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none">■ HASP HL Time■ HASP HL NetTime■ HASP SL
Protection Method	<ul style="list-style-type: none">■ Envelope-based automatic implementation■ API-based automatic implementation

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 Sentinel HASP RUS.

- *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

- 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.
- Select your protection method:
 - ◆ *Envelope-based automatic implementation*
Envelope the executable file using Sentinel HASP Envelope, specifying its Feature ID.
 - ◆ *API-based automatic implementation*
In your code, insert a Sentinel HASP Run-time 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 HASP Envelope-based or Sentinel HASP Run-time API-based automatic implementation.

- In Sentinel HASP Business Studio, create a Product that includes the Feature ID and define an expiration date or the number of days until expiration of Phase 1.
- Distribute your software with the appropriate Sentinel HASP protection key programmed with the license.
- Subject to receiving payment for Phase 2 from the user, extend the license remotely using Sentinel HASP RUS.

Micro-rental

Sentinel HASP Functionality	Manages the time period over which your software can be used
Software Distribution Method	<ul style="list-style-type: none"> ■ Physical package ■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none"> ■ HASP HL Time ■ HASP HL NetTime ■ HASP SL
Protection Method	<ul style="list-style-type: none"> ■ 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

- Select the executable file that you want to license, and determine by which Feature ID it will be identified.
- In your code, insert a Sentinel HASP Run-time API Login call to the Feature ID.
- 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.
- In Sentinel HASP Business Studio, in the key memory, define the total number of software activity hours that has been purchased.
- Envelope your software for additional security (optional).
- Distribute your software with the appropriate Sentinel HASP protection key programmed with the license.
- Using the Sentinel HASP Run-time API and the key’s built-in clock:
 1. Calculate the accumulated active time.
 2. Write the result to the key memory.
 3. Verify that the accumulated time has not exceeded the number of purchased hours.
 4. When the number of purchased hours is about to expire, display a warning message.
- When payment is received for additional usage, renew the license remotely using Sentinel HASP RUS.

Subscription

Sentinel HASP Functionality	Creates an unconditional license that can be updated remotely
Software Distribution Method	<ul style="list-style-type: none"> ■ Physical package ■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none"> ■ HASP HL Time ■ HASP HL NetTime ■ HASP SL
Protection Method	<ul style="list-style-type: none"> ■ Envelope-based automatic implementation ■ API-based automatic implementation

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

- Select the executable file that you want to license, and determine by which Feature ID it will be identified.
- Select the protection method for your software:
 - ◆ *Envelope-based automatic implementation*
Envelope the executable file using Sentinel HASP Envelope, specifying its Feature ID.
 - ◆ *API-based automatic implementation*
In your code, insert a Sentinel HASP Run-time API Login call to the Feature ID.
- In Sentinel HASP Business Studio, create a Product that includes the Feature ID for your initial software and define a perpetual license for the Feature.
- 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 HASP Envelope or Sentinel HASP Run-time API-based).

- In Sentinel HASP Business Studio, create a Product that includes the Feature ID for the update-installation component and define an expiration date for that Feature.
- Envelope your software for additional security (optional).
- Distribute your software with the appropriate Sentinel HASP protection key programmed with the license.
- During the subscription period, use Sentinel HASP RUS to send updates to the subscriber. The updates are handled by the update-installation component in your software. Optionally, use Sentinel HASP to encrypt the update files so that the HASP protection key is required to decrypt them.
- Continue sending updates as long as the end user's subscription is valid.
- When the end user renews the subscription, use Sentinel HASP RUS to update the expiration date for the update-installation component's license.

Pay-by-Peak Time (Peak Time)

Sentinel HASP Functionality	Compares a value in the key memory with a value collected during run-time
Software Distribution Method	<ul style="list-style-type: none"> ■ Physical package ■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none"> ■ HASP HL Time ■ HASP HL NetTime ■ HASP SL
Protection Method	<ul style="list-style-type: none"> ■ 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

- Select the executable file that you want to license, and determine by which Feature ID it will be identified.
- In your code, insert a Sentinel HASP Run-time API Login call to the Feature ID.
- Determine what constitutes “active” for the purpose of calculating 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.
- In Sentinel HASP Business Studio, in the 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).
- Envelope your software for additional security (optional).
- Distribute your software with the appropriate Sentinel HASP protection key programmed with the license.
- Using the Sentinel HASP Run-time API and the key’s built-in clock:
 1. Calculate the accumulated active time for each separate rate.
 2. Calculate the total number of “usage units” consumed.
 3. Write the result to the key memory.
 4. Verify that the accumulated consumption has not exceeded the total number of “usage units” defined in the key memory.
 5. When the “usage units” are about to expire, display a warning message.
- Using Sentinel HASP RUS, replenish the pool of “usage units” when the license is renewed.

Time-based Overdraft

Sentinel HASP Functionality	Manages the time period over which software can be used
Software Distribution Method	<ul style="list-style-type: none">■ Physical package■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none">■ HASP HL Time■ HASP HL NetTime■ HASP SL
Protection Method	<ul style="list-style-type: none">■ 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

- Select the executable file that you want to license, and determine by which Feature ID it will be identified.
- In your code, insert a Sentinel HASP Run-time API Login call to the Feature ID.
- In Sentinel HASP Business Studio, 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.
- Envelope your software for additional security (optional).
- Distribute your software with the appropriate Sentinel HASP protection key programmed with the license.

- Using the Sentinel HASP Run-time API and the key's built-in clock:
 - ◆ Calculate the time period.
 - ◆ 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.
- When the end user renews the license, billing includes payment for the overdraft usage in addition to the license renewal.
- After payment has been received, renew the license remotely using Sentinel HASP RUS.

Standard Counter

Sentinel HASP Functionality	Manages the maximum number of software executions
Software Distribution Method	<ul style="list-style-type: none"> ■ Physical package ■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none"> ■ All HASP HL keys except HASP HL Basic ■ HASP SL
Protection Method	<ul style="list-style-type: none"> ■ Envelope-based automatic implementation ■ API-based automatic implementation

Description

The end user purchases a predefined number of software executions, which can be defined for your software or for specific functionality. A counter-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 HASP Admin Control Center, and can order a license renewal before the license expires. The license renewal is implemented using Sentinel HASP RUS.

Implementation

- 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.
- Select your protection method:
 - ◆ *Envelope-based automatic implementation*
Envelope the executable file using Sentinel HASP Envelope, specifying its Feature ID.
 - ◆ *API-based automatic implementation*
In your code, insert a Sentinel HASP Run-time 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 HASP Envelope-based or Sentinel HASP Run-time API-based automatic implementation.

- In Sentinel HASP Business Studio, create a Product that includes the Feature ID and define the number of executions.
- Distribute your software with the appropriate Sentinel HASP protection key programmed with the license.
- Renew the license remotely using Sentinel HASP RUS.

Phased Counter

Sentinel HASP Functionality	Manages the maximum number of software executions
Software Distribution Method	<ul style="list-style-type: none">■ Physical package■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none">■ All HASP HL keys except HASP HL Basic■ HASP SL
Protection Method	<ul style="list-style-type: none">■ Envelope-based automatic implementation■ API-based automatic implementation

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 Sentinel HASP RUS.

- **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

- 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.
- Select your protection method:
 - ◆ *Envelope-based automatic implementation*
Envelope the executable file using Sentinel HASP Envelope, specifying its Feature ID.
 - ◆ *API-based automatic implementation*
In your code, insert a Sentinel HASP Run-time 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 HASP Envelope-based or Sentinel HASP Run-time API-based automatic implementation.

- In Sentinel HASP Business Studio, create a Product that includes the Feature ID and define the number of executions included in Phase 1.
- Distribute your software with the appropriate Sentinel HASP protection key programmed with the license.
- Subject to receiving payment for Phase 2 from the end user, replenish the number of executions remotely using Sentinel HASP RUS.

Capacity (CPU/Memory/Disk)

Sentinel HASP Functionality	Manages resource usage
Software Distribution Method	<ul style="list-style-type: none">■ Physical package■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none">■ All HASP HL keys except HASP HL Basic■ HASP SL
Protection Method	<ul style="list-style-type: none">■ 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

- Select the executable file that you want to license, and determine by which Feature ID it will be identified.
- In your code, insert a Sentinel HASP Run-time API Login call to the Feature ID.
- 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.
- In Sentinel HASP Business Studio, create a Product that includes the Feature ID and define the license terms—for instance, a perpetual license or a time-limited license.
- In Sentinel HASP Business Studio, in the key memory, define the capacity that has been purchased.
- Envelope your software for additional security (optional).
- Distribute your software with the appropriate Sentinel HASP protection key programmed with the license.

- Using the Sentinel HASP Run-time API:
 1. Calculate the accumulated usage.
 2. Write the result to the key memory.
 3. Verify that the accumulated usage has not exceeded the purchased capacity.
 4. When purchased capacity has almost expired, display a warning message.
- When payment is received for additional usage, renew the license remotely using Sentinel HASP RUS.

Locked Licenses

A locked license is limited to usage on a specific machine or by a specific end user.

The locked licensing models described below are:

- *Machine-locked*
- *User-locked*

Machine-locked

Sentinel HASP Functionality	Creates an activation key that is locked to a specific machine
Software Distribution Method	<ul style="list-style-type: none"> ■ Physical package ■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none"> ■ All HASP HL keys except HASP HL Basic ■ HASP SL
Protection Method	<ul style="list-style-type: none"> ■ Envelope-based automatic implementation ■ API-based automatic implementation

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 HASP SL key

This model is applicable when a HASP SL key provides sufficient security for your needs.

- Select and implement the required licensing model.
- Distribute your software using a HASP SL key. HASP SL keys are always locked to a specific machine.

Implementation 2—Combined locking to both a HASP SL key and a HASP HL key

This model is applicable when you want to lock your software to a HASP HL key for enhanced security, and also wants to use a HASP SL key to lock your software to a specific machine. The HASP SL key will require remote activation.

- 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 HASP HL key, and the other to lock the license to the HASP SL key and the machine.
- Select your protection method:
 - ◆ *For combined Envelope-based and API-based automatic implementation*
Envelope the executable file using Sentinel HASP Envelope, specifying one of the Feature IDs. In your code, insert a Sentinel HASP Run-time API Login call to other Feature ID.
 - ◆ *For API-based automatic implementation*
In your code, insert Sentinel HASP Run-time API Login calls to both Feature IDs.
- In Sentinel HASP Business Studio, create two Products, one for each Feature ID. Define the license terms for both Products—for example, a counter-based license or a time-limited license.
- Burn a HASP HL key for one of the Products and create a HASP SL Product Key for the other Product.
- Distribute your software with both Sentinel HASP protection keys.

Implementation 3—Locking to a HASP HL key

This model is applicable when you want to lock the license to both a machine and a HASP HL key—but for security reasons, the end user will not be able to activate a HASP 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 key, and the run-time identifiers are written to the Read/Write memory on the HASP HL key and validated against the initial identifiers.

Note:

It is recommended that you contact Sentinel HASP Professional Services for a detailed implementation plan.

User-locked

Sentinel HASP Functionality	Compares end user data saved in the key memory with a value collected during run-time
Software Distribution Method	<ul style="list-style-type: none"> ■ Physical package ■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none"> ■ All HASP HL keys except HASP HL Basic ■ HASP SL
Protection Method	<ul style="list-style-type: none"> ■ 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 HASP 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.

 - ◆ When a license is purchased, request the login user name of the end user for whom the license is intended.
 - ◆ Use Sentinel HASP Business Studio to save the user name to the Read-Only memory of a Sentinel HASP protection key.
 - ◆ During run-time, read the user name from the machine, and use the Sentinel HASP Run-time API to validate it against the user name saved on the Sentinel HASP 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 HASP HL key is used, your software can be installed on more than one computer, but can only be accessed when the HASP HL key is connected.

 - ◆ During installation, request the end user to define a user name and password.
 - ◆ Use the Sentinel HASP Run-time API to save the data to the Read/Write memory on the HASP HL key.
 - ◆ During run-time, require the end user to log in, and validate the user name and password against the data saved on the Sentinel HASP protection key.

Mobile Licenses

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 mobile licensing models described below are:

- *Portable*
- *Commuter*
- *Software on a Key*

Portable

Sentinel HASP Functionality	Locks the license to a hardware-based HASP HL key
Software Distribution Method	■ Physical package
Applicable Key Types	■ All HASP HL keys
Protection Method	■ Envelope-based automatic implementation ■ API-based automatic implementation

Description

Your software can be installed on any number of machines, providing flexibility, but can only run on the machine to which the HASP HL key is connected.

Implementation

- Select and implement the required licensing model.
- Distribute your software with the appropriate HASP HL key, programmed with the license.

Commuter

Sentinel HASP Functionality	Enables a network-based license to be detached to a separate machine while locked to a HASP SL key
Software Distribution Method	■ Electronic distribution
Applicable Key Types	■ HASP SL Net
Protection Method	■ Envelope-based automatic implementation ■ API-based automatic implementation

Description

A license can be temporarily detached from a network pool—using Sentinel HASP 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

- Select and implement the network concurrency licensing model, ensuring that the license can be locked to a HASP SL key and that detachable licenses are enabled.
- Distribute your software with a HASP SL key, ensuring that the system administrator at your end-user site knows how to permit and manage detachable licenses.
- 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

Sentinel HASP Functionality	Locks the license to a HASP HL Drive key that also contains your software
Software Distribution Method	■ Physical package
Applicable Key Types	■ HASP HL Drive
Protection Method	■ Envelope-based automatic implementation ■ API-based automatic implementation

Description

Both your software and the license are stored on a HASP HL Drive key, providing maximal mobility. The HASP HL Drive key contains 2 GB or 4 GB of flash memory 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 license models for which a hardware-based key is used, except those requiring HASP HL Time, HASP HL NetTime or HASP HL Net keys.

Implementation

- Select and implement the required licensing model.
- Distribute your software on a HASP HL Drive key, together with the software's license.

Network Licensing 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 HASP 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 models such as component-based or metering. A network license can be concurrency-based, site-specific, or both.

The network licensing models described below are:

- *Limited Concurrent End Users in a Network*
- *Time-limited Concurrent End Users in a Network*
- *Execution-limited Concurrent End Users in a Network*
- *Volume*
- *Site*

Limited Concurrent End Users in a Network

Sentinel HASP Functionality	Manages the number of concurrent software end users
Software Distribution Method	<ul style="list-style-type: none">■ Physical package■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none">■ HASP HL Net■ HASP HL NetTime■ HASP SL
Protection Method	<ul style="list-style-type: none">■ Envelope-based automatic implementation■ API-based automatic implementation

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 HASP Admin Control Center provides the end users' system administrator with the tools to track license users, and to terminate an inactive session.

Implementation

- Select the executable file that you want to license, and determine by which Feature ID the file or function will be identified.
- Select your protection method:
 - ◆ *Envelope-based automatic implementation*
Envelope the executable file using Sentinel HASP Envelope, specifying its Feature ID.
 - ◆ *API-based automatic implementation*
In your code, insert a Sentinel HASP Run-time API Login call to the Feature ID.
- In Sentinel HASP Business Studio:
 1. Create a Product that includes the Feature ID, and define the license type as Perpetual.
 2. 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.
- Distribute your software with a Sentinel HASP protection key programmed with the license.

Time-limited Concurrent End Users in a Network

Sentinel HASP Functionality	Manages the number of concurrent software end users in a network and the time period over which your software can be used
Software Distribution Method	<ul style="list-style-type: none"> ■ Physical package ■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none"> ■ HASP HL NetTime ■ HASP SL
Protection Method	<ul style="list-style-type: none"> ■ Envelope-based automatic implementation ■ API-based automatic implementation

Description

A combined concurrency- 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 HASP Admin Control Center provides the end user's system administrator with the tools to track license users, and to terminate an unused session.

Implementation

- Select the executable file that you want to license, and determine by which Feature ID it will be identified.
- Select your protection method:
 - ◆ *Envelope-based automatic implementation*
Envelope the executable file using Sentinel HASP Envelope, specifying its Feature ID.
 - ◆ *API-based automatic implementation*
In your code, insert a Sentinel HASP Run-time API Login call to the Feature ID.
- In Sentinel HASP Business Studio:
 1. Create a Product that includes the Feature ID, and define the expiration date or number of days until expiration.
 2. 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.
- Distribute your software with the appropriate network-based Sentinel HASP protection key programmed with the license.

Execution-limited Concurrent End Users in a Network

Sentinel HASP Functionality	Manages the number of concurrent software end users in a network and the maximum number of software executions
Software Distribution Method	<ul style="list-style-type: none"> ■ Physical package ■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none"> ■ HASP HL Net ■ HASP HL NetTime ■ HASP SL
Protection Method	<ul style="list-style-type: none"> ■ Envelope-based automatic implementation ■ API-based automatic implementation

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 HASP Admin Control Center provides the end users' system administrator with the tools to track license users, and to terminate an unused session.

Implementation

- 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.
- Select your protection method:
 - ◆ *Envelope-based automatic implementation*
Envelope the executable file using Sentinel HASP Envelope, specifying its Feature ID.

◆ *API-based automatic implementation*

In your code, insert a Sentinel HASP Run-time 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 HASP Envelope-based or Sentinel HASP Run-time API-based automatic implementation.

- In Sentinel HASP Business Studio:
 1. Create a Product that includes the Feature ID, and define the maximum number of executions.
 2. 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.
- Distribute your software with the appropriate network-based Sentinel HASP protection key programmed with the license.

Volume

Sentinel HASP Functionality	Enables a network-based license to be detached to a separate machine while locked to a HASP SL key
Software Distribution Method	■ Electronic distribution
Applicable Key Types	■ HASP SL
Protection Method	■ Envelope-based automatic implementation ■ API-based automatic implementation

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

- Select the executable file that you want to license, and determine by which Feature ID it will be identified.
- Select your protection method:
 - ◆ *Envelope-based automatic implementation*
Envelope the executable file using Sentinel HASP Envelope, specifying its Feature ID.
 - ◆ *API-based automatic implementation*
In your code, insert a Sentinel HASP Run-time API Login call to the Feature ID.
- In Sentinel HASP Business Studio, 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 HASP SL key, and detachable licenses.
- Distribute your software with a HASP SL key for network use, ensuring that the system administrator at your end-user site knows how to permit and manage detachable licenses.
- Using the Sentinel HASP Run-time 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

Sentinel HASP Functionality	Locks the license to a specific domain, network, or subnet
Software Distribution Method	<ul style="list-style-type: none">■ Physical package■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none">■ All HASP HL keys except HASP HL Basic■ HASP SL
Protection Method	<ul style="list-style-type: none">■ 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 models in this guide.

Implementation

- Select and implement the required licensing model.
- Envelope your software for additional security (optional).
- Distribute your software using the appropriate Sentinel HASP protection key.
- To lock the Sentinel HASP protection key to the license, collect the site identifier (domain, subnet or network) from the customer. An identification value is written to the Sentinel HASP protection key. The application then validates the identifier every time your software runs.
- There are two ways in which you can collect site-specific data and save it on the Sentinel HASP 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 HASP Business Studio to save the identification value to the Read-Only memory of the Sentinel HASP 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 HASP Run-time API to verify that there is no existing site identifier saved in the Read/Write memory on the Sentinel HASP protection key.

If the memory does not contain an existing site identifier, save the value to the Read/Write memory on the Sentinel HASP protection key.

- During run-time, read the site identifier, and use Sentinel HASP Run-time API to validate it against the identification value saved on the Sentinel HASP protection key.

Sales Boosting Licensing Models

The sales boosting licensing models described below are:

- *KickStart (Quick-delivery Grace)*
- *Referral-based Sales*
- *Automatic Sales Agent*
- *Reselling Via Distribution Channels*

KickStart (Quick-delivery Grace)

Sentinel HASP Functionality	Grants a grace period to use software until key is delivered
Software Distribution Method	<ul style="list-style-type: none"> ■ Physical package ■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none"> ■ All HASP HL keys except HASP HL Basic ■ HASP SL
Protection Method	<ul style="list-style-type: none"> ■ Envelope-based automatic implementation ■ API-based automatic implementation

Description

Locking a license to a HASP HL key provides a higher level of security than locking to a HASP SL key, but delivery of the HASP 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 HASP 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 can be defined for any period between 1 and 90 days.

- **Phase 2:** The HASP 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 Sentinel HASP RUS.

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

- Determine which global Feature ID you will use for the KickStart license.
- 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.
- Select a protection method and do one of the following:
For Envelope-based automatic implementation:
 - ◆ Determine which global Feature ID you will use for the full license.
 - ◆ Create two executable files, one with limited functionality for the KickStart license, and the other with full functionality for the full license.

- ◆ Envelope each executable file separately, using the global Feature IDs you defined for the KickStart and full licenses respectively.

For API-based automatic implementation:

- ◆ In your code, insert a Sentinel HASP Run-time API Login call to the global Feature ID for the KickStart license.
- ◆ In your code, for each software function you want to include only in the full license, insert Sentinel HASP Run-time API Login calls to the appropriate Feature IDs.
- In Sentinel HASP Business Studio:
 - ◆ Create a Product that includes the global Feature ID for the KickStart license.
 - ◆ Select the Provisional product attribute.
- Distribute your software with Sentinel HASP 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.
- In Sentinel HASP Business Studio:
 1. Create a Product that includes the full license Feature IDs.
 2. 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.

- Distribute your software with a Sentinel HASP protection key programmed with the full license.

Referral-based Sales

Sentinel HASP Functionality	Sentinel HASP protection
Software Distribution Method	<ul style="list-style-type: none"> ■ Physical package ■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none"> ■ All HASP HL keys ■ HASP SL
Protection Method	<ul style="list-style-type: none"> ■ Envelope-based automatic implementation ■ API-based automatic implementation

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 and or on a website.
- **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 effect the referral-based sales model, based on:

- Using trialware as the evaluation mechanism.
- Distributing the purchased software with a software-based HASP SL key.
- Collecting information from the referee during software activation.

Implementation

- Create a trialware version of your software.

- End users who have already purchased your software send the trialware to other potential users.
- When a new user purchases your software—as part of your software activation process using Sentinel HASP functionality—prompt the new user to provide you with the name and contact information of the end user who referred your software to them.
- Reward the referrer.

Note:

This is a typical implementation, however, the referral-based sales model can also be applied to other licensing models, including those models that use a hardware-based HASP HL key.

Automatic Sales Agent

Sentinel HASP Functionality	Manages module usage
Software Distribution Method	<ul style="list-style-type: none">■ Physical package■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none">■ All HASP HL keys except HASP HL Basic■ HASP SL
Protection Method	<ul style="list-style-type: none">■ 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 HASP, 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 HASP 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 Sentinel HASP RUS, and the relevant bonus modules are changed to fully paid modules.

Implementation

- Determine which Feature ID you will use for global protection of your software.
- Select the modules that you want to license separately, and determine by which Feature ID each of the modules will be identified.
- In your code, insert Sentinel HASP Run-time API Login calls to all Feature IDs.
- In Sentinel HASP Business Studio, create a Product that includes only the global software Feature ID and define the license terms.
- 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
- In Sentinel HASP Business Studio, in the key memory, define the usage threshold.
- Envelope your software for additional security (optional).
- Distribute your software with the appropriate Sentinel HASP protection key programmed with the license for the initial purchase, not including licenses for the bonus modules.
- Using the Sentinel HASP Run-time API:
 1. Calculate the accumulated usage of the gauging parameters.
 2. Write the result to the key memory.
 3. 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
- In Sentinel HASP Business Studio, 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.
- When the end user decides to purchase a license for a bonus module, update the license on the Sentinel HASP protection key to include the purchased module, using Sentinel HASP RUS.

Reselling Via Distribution Channels

Sentinel HASP Functionality	Employs a distribution channel and the Sentinel HASP Business Studio GUI to manage two-tier distribution
Software Distribution Method	<ul style="list-style-type: none"> ■ Physical package ■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none"> ■ All HASP HL keys except HASP HL Basic ■ HASP SL
Protection Method	<ul style="list-style-type: none"> ■ Envelope-based automatic implementation ■ API-based automatic implementation

Description

A two-tier distribution method that enables Value Added Resellers (VARs) to employ your Master key, Batch Code, and Sentinel HASP database to manage the handling of orders and production of licenses.

The VAR can enter orders and sell your software directly to end users, using the licenses stored on your Master key. Two-tier distribution can be implemented for reselling any of the licensing models in this guide.

Implementation

- Using Sentinel HASP Business Studio:
 1. Create a Distribution Channel for each VAR.
 2. Define the Products for each VAR, and associate each Product with the relevant Distribution Channel. You can associate a Product with multiple Distribution Channels.
 3. Define one or more Business Studio *Distribution Channel* user accounts for each VAR. Associate each user account with the relevant Distribution Channel. Provide each VAR with the user names and passwords for their accounts.
- The VAR should install Business Studio (part of the Vendor Suite) and connect to your Sentinel HASP database. (There is no need for the VAR to install Business Studio Server.) When the VAR logs in to Business Studio, they will be able to see only the Products, customers, and orders associated with their Distribution Channel. The VAR will be able to define customers, enter customer orders for the accessible Products, and generate end-user licenses. The licenses will be taken from the pool on your Master key.

Note:

This model is based on a trust relationship with the reseller - no limit is set on the number of licenses that the reseller can produce. You can also use a quota-based model in which you provide the reseller with a HASP key (referred to as the *VAR key*) that you dedicate for the purpose of containing a quota of licenses. The reseller can produce licenses for resale up to the number of licenses in the quota. Order definition and management can be handled as described above. However, a quota-based model requires you to develop a customized client application for order production. The client connects to the Business Studio server to produce the license and consumes the corresponding licenses from the local VAR key. For more information, contact Technical Support.

Perpetual Licensing model

Sentinel HASP Functionality	Creates an unconditional license
Software Distribution Method	<ul style="list-style-type: none"> ■ Physical package ■ Electronic distribution
Applicable Key Types	<ul style="list-style-type: none"> ■ All HASP HL keys ■ HASP SL
Protection Method	<ul style="list-style-type: none"> ■ Envelope-based automatic implementation ■ API-based automatic implementation

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. More sophisticated licensing models are implemented with future releases.
- A limited license (“bronze”) is converted to a perpetual license (“gold”) for additional payment.

Implementation

- Select the executable file that you want to license, and determine by which Feature ID it will be identified.
- Select your protection method:
 - ◆ *Envelope-based automatic implementation*
Envelope the executable file using Sentinel HASP Envelope, specifying its Feature ID.
 - ◆ *API-based automatic implementation*
In your code, insert a Sentinel HASP Run-time API Login call to the Feature ID.
- In Sentinel HASP Business Studio, create a Product that includes the Feature ID and define a perpetual license for the Feature.
- Use Sentinel HASP RUS to update a license currently held by the end user with the new license.

Part 6 Appendices

In this section:

- **Appendix A: Troubleshooting**
Provides a checklist to help you solve some of the most common problems that your customers might encounter when using the HASP HL keys. Also includes a list of specific problems you or your customers may experience, together with the solutions.
- **Appendix B: Sentinel HASP Run-time API Reference**
Provides an overview of the functions that make up the Sentinel HASP Run-time API. Also includes structural declarations and detailed information on individual Sentinel HASP Run-time API functions, and a summary and description of all API return codes.
- **Appendix C: Sentinel HASP Run-time Network Activity**
Describes the type of network activity that occurs in the communication between the HASP License Manager and a protected application, and between the local HASP License Manager and remote HASP License Managers.
- **Appendix D: How Sentinel HASP Detects Machine Cloning**
Describes the techniques employed by Sentinel HASP to prevent unauthorized use of protected software when the virtual machine on which the software is installed is cloned.
- **Appendix E: How Sentinel HASP Protects Time-based Licenses Locked to HASP SL Keys**
Describes the technology used in Sentinel HASP to prevent a user from extending the duration of a software license that is locked to a HASP SL key.
- **Appendix F: Understanding the Sentinel HASP Master Key Licenses**
Describes the SafeNet Sentinel HASP licensing model for software vendors.

- **Appendix G: Glossary**

Provides a glossary of Sentinel HASP-related terminology.

Appendix A

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 HASP HL keys. The second part lists specific problems you or your customers may experience, together with the solutions.

HASP HL keys conform to the highest standards of quality assurance. However, like any other PC peripheral device, a HASP 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:

<http://www.safenet-inc.com/technicalsupport.aspx>

The Knowledge Base contains a comprehensive listing of solutions to general and specific problems.

To avoid potential difficulties, ensure you are using current Sentinel HASP software versions. Contact your local SafeNet representative for the latest updates, or visit the SafeNet international downloads page at: <http://www3.safenet-inc.com/download>

Checklist

If a customer reports a problem, check the following:

- What the returned error code or message says. For additional information, see [API Status Codes on page 328](#).
- Whether a HASP 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, and so on.
- Whether Sentinel HASP Admin Control Center can access the HASP HL key.
- Whether the problem occurs when the protected application runs on another PC of the same model.

Problems and Solutions

Problem	HASP HL key drivers do not install.
Solution	Are older HASP 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 HASP HL key driver documentation. After the older drivers are removed, install the HASP HL drivers. For additional information, see the <i>Sentinel HASP Installation Guide</i> .

Problem	You receive an error message when using <code>haspdinst.exe</code> to install the HASP HL key driver under Windows 2000/XP/2003/Vista.
Solution	Review the <code>haspdinst.exe</code> installation instructions. Alternatively, try to install the drivers using the <code>HASPUserSetup.exe</code> . For additional information, see the <i>Sentinel HASP Installation Guide</i> .

Problem	The protected application cannot find a HASP HL key.
Solution	<ul style="list-style-type: none"> ■ Does the HASP HL key LED light up? If not, this could be for one of the following reasons: <ol style="list-style-type: none"> 1. 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. 2. The required HASP HL key drivers are not installed. If you are running Sentinel HASP on a Windows platform, check for an entry for Sentinel HASP in the Device Manager utility. If there is no entry, you must install the drivers using one of the methods in the <i>Sentinel HASP Installation Guide</i>. 3. Check if the USB port is functioning correctly. Disconnect all other USB devices from their respective ports. Connect the HASP HL key to a different USB port. Try using a different USB device in the port from which the HASP HL key was not accessible. ■ Open the Windows Services window and check that HASP License Manager is running. ■ Check that the Batch Code on the HASP HL key matches the Batch Code of the protected application.

Problem	The application takes a long time to find the Sentinel HASP protection key on a large network.
Solution	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 HASP protection key at a specific address, which is much faster.

Problem	You receive an error message indicating that HASP License Manager was not found.
Solution	<p>The error message might be for one of the following reasons:</p> <ul style="list-style-type: none"> ■ HASP License Manager was not loaded. Try restarting HASP License Manager in the Windows Services window. ■ There is a communication error with the machine on which the Sentinel HASP protection key is located. If you repeatedly receive the error message, try using a different search mechanism.

Problem	You cannot add files when using the DataHASP utility.
Solution	<p>The problem may occur for one of the following reasons:</p> <ul style="list-style-type: none">■ You are attempting to add a list that includes problematic files. Remove all problematic files, that are marked in red in the File list.■ You are attempting to add a file that is outside the scope of the filters defined in Sentinel HASP Envelope. You must protect your software again using the new file filter settings. <p>For additional information about working with the DataHASP utility, see Working with the DataHASP Encryption Utility on page 109.</p>

Problem	When using DataHASP, you receive a message that no data filters were defined for a program in a Sentinel HASP Envelope project.
Solution	The problem cannot be solved using the DataHASP utility. You need to use Sentinel HASP Envelope to protect your software again, and to specify file filter settings.

Problem	When generating scheduled reports using non-Latin characters (such as Japanese or Chinese), characters are not displayed correctly.
Solution	Configure the Sentinel HASP Business Studio Server to use an appropriate TrueType font that supports the required character set. For more information, see the <i>Sentinel HASP Installation Guide</i> .

Appendix B

Sentinel HASP Run-time API Reference

In this appendix:

- Introduction
- API Function Overview
- Detailed Description of Functions
- API Status Codes

Introduction

The Sentinel HASP Run-time API enables you to develop applications that incorporate functions to manage protection and licensing of protected software on end user computers.

This appendix provides a basic description of the available API functions.

For a description of using the API to protect your software, see Chapter 3, *Sentinel HASP Run-time API Protection*.

For the complete set of documentation for the Sentinel HASP Run-time API, see the Help files under:

...\Program Files\SafeNet\Sentinel HASP\API\Runtime\

Sentinel HASP ToolBox

To assist you to fully understand the functionality of each API call, use Sentinel HASP ToolBox. This tool enables you to:

- Test function calls.
- Understand the required parameters.
- Anticipate return values.

Sentinel HASP ToolBox can be accessed from the Sentinel HASP Vendor Suite.

API Samples

Each Sentinel HASP installation includes API samples for various programming languages. Use these samples to integrate Sentinel HASP protection into your own code.

Every sample folder includes a Sentinel HASP header file. Refer to the SafeNet, Inc. Website and the Sentinel HASP installation DVD for information on available sample programs for specific programming languages.

API Function Overview

The Sentinel HASP Run-time API functions in this reference apply to the C programming language interface.

The information for each function includes the following:

- **Name:** The function name, as specified in the C language interface.
- **Description:** A brief description of the main purpose of the function.
- **Syntax:** The specific function declaration in the C programming language.
- **Parameters:** The parameters for the function.

- **Returns:** All the possible returns associated with the execution of the function.
- **Usage Notes:** Detailed information on how to use the function.

Note:

The information in this Appendix is written for the Sentinel HASP Run-time API. If you are using an older version of the HASP API, refer to the appropriate API documentation for the version that you are using. You can download the relevant document from the SafeNet website.

The following table lists the available Sentinel HASP Run-time API functions.

Function	Description
<code>hasp_datetime_to_hasptime()</code>	Converts a date and time value to hasptime
<code>hasp_decrypt()</code>	Decrypts a buffer using the AES encryption algorithm
<code>hasp_detach()</code>	Detaches a Product and its license from a HASP SL key, according to customizable parameters
<code>hasp_encrypt()</code>	Encrypts a buffer using the AES encryption algorithm
<code>hasp_free()</code>	Releases allocated memory resources
<code>hasp_get_rtc()</code>	Reads the current time from a HASP HL Time or HASP HL NetTime key
<code>hasp_get_sessioninfo()</code>	Retrieves information regarding a session context
<code>hasp_get_info()</code>	Retrieves information according to customizable search parameters, and presents it according to customizable formats
<code>hasp_get_size()</code>	Retrieves the byte size of a memory file from a Sentinel HASP protection key
<code>hasp_hasptime_to_datetime()</code>	Converts a time value into a date and time
<code>hasp_login()</code>	Logs in to a Feature, establishing a session context
<code>hasp_login_scope()</code>	Retrieves login information according to customizable search parameters
<code>hasp_logout()</code>	Logs out from a context or session

Function	Description
<code>hasp_read()</code>	Reads the memory of a Sentinel HASP protection key
<code>hasp_update()</code>	Writes an update for a HASP license
<code>hasp_write()</code>	Writes to the memory of a Sentinel HASP protection key

The topic *[Blinking the HASP HL Key LED on page 331](#)* describes how you can implement functionality in your custom application to cause a HASP HL key attached to a user's computer to blink on and off.

Detailed Description of Functions

This section provides a detailed description of each Sentinel HASP Run-time API function, using the format described in the preceding section.

hasp_datetime_to_hasptime()

Description

Converts a date and time value to hasptime (the number of elapsed seconds since 01/01/1970).

Syntax

```
hasp_status_t HASP_CALLCONV hasp_datetime_to_hasptime(  
  
    unsigned int  day,  
    unsigned int  month,  
    unsigned int  year,  
    unsigned int  hour,  
    unsigned int  minute,  
    unsigned int  second,  
    hasp_time_t * time  
)
```

Parameters

day	Input for day value (range 1-31)
month	Input for month value (range 1-12)
year	Input for year value (range 1970+)
hour	Input for hour value (range 0-23)
minute	Input for minute value (range 0-59)
second	Input for second value (range 0-59)
time	Pointer to the resulting time value

Return Values

HASP_STATUS_OK	Request was successfully completed
HASP_INV_TIME	Passed time value is outside the supported value range

Usage Notes

The converted date and time value reflects the number of elapsed seconds since January 1, 1970. This conversion function is used in conjunction with the API functions that set or retrieve values for the real-time clock (RTC) in the HASP HL Time, HASP HL NetTime and HASP SL keys.

Related Topics

- `hasp_encrypt()`
- `hasp_hasptime_to_datetime()`

hasp_decrypt()

Description

Reverses the operation of the `hasp_encrypt()` function applied on a data buffer, returning the data to its unencrypted state.

Syntax

```
hasp_status_t HASP_CALLCONV hasp_decrypt(  
    hasp_handle_t handle,  
    void * buffer,  
    hasp_size_t length  
)
```

Parameters

handle	Handle for the session
buffer	Pointer to the buffer to be decrypted
length	Size (in bytes) of the buffer to be decrypted—16 bytes minimum required

Return Values

HASP_STATUS_OK	Request was successfully completed
HASP_INV_HND	Invalid input handle
HASP_DEVICE_ERR	Input/output error in HASP SL secure storage, OR, in the case of a HASP HL key, USB communication error
HASP_TIME_ERR	System time has been tampered with
HASP_TOO_SHORT	Encryption data length is too short
HASP_SCHAN_ERR	Communications error in secure channel
HASP_ENC_NOT_SUPP	Hardware does not support encryption type

HASP_BROKEN_SESSION	Session was interrupted
HASP_LOCAL_COMM_ERR	Communication error occurred between the application and the local HASP License Manager
HASP_REMOTE_COMM_ERR	Communication error occurred between the local and remote HASP License Managers

Usage Notes

Decrypts data using the encryption engine in the Sentinel HASP protection key. The specific session handle determines which Sentinel HASP protection key and which Feature ID encrypts the data buffer. The encryption key remains in the Sentinel HASP protection key. If the decryption fails, the data buffer is not modified.

Related Topics

`hasp_encrypt()`

hasp_detach()

Description

Performs either of the following functions:

- Detaches a Product and its license from a HASP SL key (on the host machine), according to customizable parameters. Features defined as detachable and memory files that belong to the Product are detached. You do not need to be logged in to a Feature in order to use this function.
- Cancels a detached Product and its license from the HASP SL key (on the recipient machine), typically before the detached Product was due to expire.

You do not need to be logged in to a Feature in order to use this function.

Syntax

```
hasp_status_t HASP_CALLCONV hasp_detach(  
    const char *detach_action,  
    const char *scope,  
    hasp_vendor_code_t vc  
    const char *recipient  
    char **info  
)
```

Parameters

detach_action	<p>For Detach: Detach operation parameters, in XML format. The following example specifies that a detached license for Product ID 12345 will expire after 604,800 seconds (one week):</p> <pre><detach> <product id="12345"> <duration>604800</duration> </product> </detach></pre> <p>For Cancel: Cancel operation parameters, in XML format. The following example specifies that the HASL SL protection key with ID "12321" for the detached license should be cancelled:</p> <pre><cancel> <hasp id="12321"> </cancel></pre>
scope	Search parameters for the Product that is to be detached. For more information, refer to Scope Input XML Tags in the <i>XML Tags</i> section of the <i>Sentinel HASP Run-time API Reference</i> in the Help documentation.
vc	Pointer to the Vendor Code
recipient	<p>For Detach: Definition in XML format of the recipient machine to which the detached Product and its license will be attached. Use either <code>hasp_get_info()</code> or <code>hasp_get_sessioninfo()</code>, together with the <code>HASP_RECIPIENT</code> specifier, to retrieve the recipient information. For more information, refer to the <i>XML Tags</i> section of the <i>Sentinel HASP Run-time API Reference</i> in the Help documentation.</p> <p>For Cancel: Ignored (set to Null)</p>
info	<p>Pointer to the information that is retrieved as XML text. This information is an:</p> <ul style="list-style-type: none"> ■ H2R file (for the Detach action), which can then be installed on the recipient machine ■ R2H file (for the Cancel action), which can then be installed on the host machine <p>The file can be installed using the <code>hasp_update</code> function (or using the Admin Control Center or RUS utility). Use the <code>hasp_free</code> function to release the pointer after use.</p>

Return Values

HASP_STATUS_OK	Request was successfully completed
HASP_INV_DETACH_ACTION	Invalid XML detach_action parameter
HASP_INV_RECIPIENT	Invalid XML recipient parameter
HASP_TOO_MANY_PRODUCTS	hasp_detach scope does not specify a unique Product
HASP_INV_PRODUCT	Invalid Product information
HASP_INSUF_MEMORY	System out of memory
HASP_DEVICE_ERROR	Input/output error occurred in secure storage area of HASP SL key OR In the case of a HASP HL key, USB communication error occurred
HASP_LOCAL_COMM_ERROR	Communication error occurred between the application and the local HASP License Manager
HASP_REMOTE_COMM_ERROR	Communication error occurred between the local and remote HASP License Managers

Usage Notes

The requisite Vendor Codes are stored in a VendorCodes directory in the system. You cannot run the function without the correct Vendor Codes being available.

Related Topics

- `hasp_get_info()`
- `hasp_get_sessioninfo()`

Additional information is also available in the “XML Tags” section of the *Sentinel HASP Run-time API Reference* in the Help documentation.

hasp_encrypt()

Description

Encrypts a buffer using the AES encryption algorithm.

Syntax

```
hasp_status_t HASP_CALLCONV hasp_encrypt(  
    hasp_handle_t handle,  
    void * buffer,  
    hasp_size_t length  
)
```

Parameters

handle	Handle for the session
buffer	Pointer to the buffer to be encrypted
length	Size (in bytes) of the buffer to be encrypted—16 bytes minimum required

Return Values

HASP_STATUS_OK	Request was successfully completed
HASP_INV_HND	Invalid input handle
HASP_DEVICE_ERR	Input/output error in HASP SL secure storage, OR, in the case of a HASP HL key, USB communication error
HASP_TIME_ERR	System time has been tampered with
HASP_TOO_SHORT	Encryption data length is too short
HASP_SCHAN_ERR	Communications error in secure channel
HASP_ENC_NOT_SUPP	Hardware does not support encryption type
HASP_HASP_NOT_FOUND	Required HASP key not found
HASP_LOCAL_COMM_ERR	Communication error occurred between the application and the local HASP License Manager
HASP_REMOTE_COMM_ERR	Communication error occurred between the local and remote HASP License Managers

Usage Notes

Encodes data using the encryption engine in the Sentinel HASP protection key. The specified session handle determines which Sentinel HASP protection key performs the encryption of the data buffer. The encryption key remains in the Sentinel HASP protection key. If the encoding operation fails, the data targeted for encryption is not affected. To decode the data buffer, use the `hasp_decrypt()` function.

Related Topics

[hasp_decrypt\(\)](#)

hasp_free()

Description

Releases memory resources utilized by other API functions.

Syntax

```
void HASP_CALLCONV hasp_free(char * info)
```

Parameters

info	Pointer to the memory resources to be released
------	--

Usage Notes

Used only in C code to release memory resources allocated to storing retrieved data from API calls using the [hasp_get_sessioninfo\(\)](#) and [hasp_update\(\)](#) functions. The function has no return values.

Related Topics

- [hasp_get_sessioninfo\(\)](#)
- [hasp_update\(\)](#)

hasp_get_rtc()

Description

Reads the current time from a HASP HL Time or HASP HL NetTime key.

Syntax

```
hasp_status_t HASP_CALLCONV hasp_get_rtc(  
    hasp_handle_t  handle,  
    hasp_time_t *  time  
)
```

Parameters

handle	Handle for the session
time	Pointer to the current time

Return Values

HASP_STATUS_OK	Request was successfully completed
HASP_INV_HND	Invalid input handle
HASP_HASP_NOT_FOUND	Required Sentinel HASP protection key not found
HASP_NO_BATTERY_POWER	Real-time clock has run out of power
HASP_NO_TIME	Real-time clock is not available
HASP_LOCAL_COMM_ERR	Communication error occurred between the application and the local HASP License Manager
HASP_REMOTE_COMM_ERR	Communication error occurred between the local and remote HASP License Managers

Usage Notes

Primarily used to obtain reliable timestamps that are independent of the system clock. Time values are returned as the number of seconds that have elapsed since Jan.-01-1970 0:00 hours UTC. Use the [hasp_hasptime_to_datetime\(\)](#) function to convert the output to UTC format. This function reads the HASP HL Net key and HASP HL NetTime key time, not the expiration date time.

hasp_get_sessioninfo()

Description

Retrieves information regarding a session context.

Syntax

```
hasp_status_t HASP_CALLCONV hasp_get_sessioninfo(  
    hasp_handle_t handle,  
    char * format,  
    char ** info  
)
```

Parameters

handle	Handle for the session
format	<p>XML definition for the type of output data structure.</p> <p>There are three format options:</p> <ol style="list-style-type: none">1. HASP_KEYINFO: For retrieving information on the Sentinel HASP protection key.2. HASP_SESSIONINFO: For retrieving information on the session.3. HASP_UPDATEINFO: For retrieving information on a license update usually contained in a C2V file. The retrieved information includes information on update counters, licenses and memory images currently available in a deployed Sentinel HASP protection key. <p>For more information, refer to the <i>XML Tags</i> section of the <i>Sentinel HASP Run-time API Reference</i> in the Help documentation.</p>
info	Pointer to the information which is retrieved as XML text

Return Values

HASP_STATUS_OK	Request was successfully completed
HASP_HASP_NOT_FOUND	Sentinel HASP protection key is no longer available
HASP_INV_HND	Invalid input handle

HASP_INV_FORMAT	Unrecognized format
HASP_INSUF_MEM	Out of memory
HASP_BROKEN_SESSION	Session has been interrupted
HASP_LOCAL_COMM_ERR	Communication error occurred between the application and the local HASP License Manager
HASP_REMOTE_COMM_ERR	Communication error occurred between the local and remote HASP License Managers
HASP_DEVICE_ERR	Input/output error in HASP SL secure storage, OR, in the case of a HASP HL key, USB communication error
HASP_TIME_ERR	System time has been tampered with

Usage Notes

When using the `HASP_UPDATEINFO` parameter, the Sentinel HASP protection key must be accessible by the local machine.

The retrieved session information applies to:

- A deployed Sentinel HASP protection key
- The current or a specific login session
- A license update

This function allocates memory for the information it retrieves.

To release allocated memory resources, use the `hasp_free()` function.

To convert a returned time value to the current date and time, use the `hasp_hasptime_to_datetime()` function.

Related Topics

- `hasp_free()`
- `hasp_get_info()`

hasp_get_info()

Description

Retrieves information about system components, according to customizable search parameters, and presents it according to customizable formats.

Syntax

```
hasp_status_t HASP_CALLCONV hasp_get_info(  
    char * scope,  
    char * format,  
    hasp_vendor_code_t vendor code,  
    char ** info  
)
```

Parameters

scope	Definition of the data that is to be searched. For more information, refer to the Scope XML Tags in the <i>XML Tags</i> section of the <i>Sentinel HASP Run-time API Reference</i> in the Help documentation.
format	Definition of the format in which the data is to be displayed. For more information, refer to the Scope XML Tags in the <i>XML Tags</i> section of the <i>Sentinel HASP Run-time API Reference</i> in the Help documentation.
vendor code	Pointer to the Vendor Code
info	Pointer to the information that is retrieved

Return Values

HASP_STATUS_OK	Request was successfully completed
HASP_INV_FORMAT	Unrecognized format
HASP_INV_SCOPE	Unrecognized scope
HASP_INSUF_MEM	Out of memory

HASP_BROKEN_SESSION	Session has been interrupted
HASP_LOCAL_COMM_ERR	Communication error has occurred between the application and the local HASP License Manager
HASP_REMOTE_COMM_ERR	Communication error has occurred between the local and remote HASP License Managers
HASP_DEVICE_ERR	Input/output error in HASP SL secure storage, OR, in the case of a HASP HL key, USB communication error
HASP_INV_VCODE	Invalid Vendor Code
HASP_UNKNOWN_VCODE	Vendor Code not recognized by API
HASP_INVALID_PARAMETER	Scope string too long (maximum length 32 KB)

Usage Notes

You do not need to be logged in to HASP Vendor Suite in order to use this function.

This function is used to specify conditions about where to search for information. In addition, it enables you to specify conditions about the format in which the retrieved information is presented. If retrieved information is appropriately formatted, it can be used as a template in the `hasp_login_scope()` function.

The requisite Vendor Codes are stored in a **VendorCodes** folder in your system. Without the correct Vendor Code, the function call cannot succeed.

Related Topics

- `hasp_get_sessioninfo()`
- `hasp_free()`

hasp_get_size()

Description

Retrieves the byte size of a memory file from a Sentinel HASP protection key.

Syntax

```
hasp_status_t HASP_CALLCONV hasp_get_size(  
    hasp_handle_t  handle,  
    hasp_fileid_t  fileid,  
    hasp_size_t *   size  
)
```

Parameters

handle	Handle for the session
fileid	Identifier for the file that is to be queried. Possible values are FILEID_RO or FILEID_RW for read only or read/write.
size	Pointer to the resulting file size

Return Values

HASP_STATUS_OK	Request was successfully completed
HASP_INV_HND	Invalid input handle
HASP_INV_FILEID	Unrecognized file identifier
HASP_TIME_ERR	System time has been tampered with
HASP_HASP_NOT_FOUND	Sentinel HASP protection key is no longer available
HASP_LOCAL_COMM_ERR	Communication error occurred between the application and the local HASP License Manager
HASP_REMOTE_COMM_ERR	Communication error occurred between the local and remote HASP License Managers

Usage Notes

This function is used to determine the file size of the Sentinel HASP protection key memory. The file size enables you to determine the largest possible byte size offset. This information is useful when reading or writing to the memory of a Sentinel HASP protection key. The first byte in a file has the index 0.

Related Topics

- `hasp_read()`
- `hasp_write()`

hasp_hasptime_to_datetime()

Description

Converts a time value (elapsed seconds since January 1, 1970) into a date and time.

Syntax

```
hasp_status_t HASP_CALLCONV hasp_hasptime_to_datetime(  
    hasp_time_t  time,  
    unsigned int * day,  
    unsigned int * month,  
    unsigned int * year,  
    unsigned int * hour,  
    unsigned int * minute,  
    unsigned int * second  
)
```

Parameters

time	Pointer for placing time value
day	Pointer for day value
month	Pointer for month value
year	Pointer for year value
hour	Pointer for hour value
minute	Pointer for minute value
second	Pointer for second value

Return Values

HASP_STATUS_OK	Request was successfully completed
HASP_INV_TIME	Passed time value is outside the supported value range

Usage Notes

All values are based on Coordinated Universal Time (UTC). The converted date and time value reflects the number of elapsed seconds since Jan. 1, 1970. This conversion function is used in conjunction with the API functions that set or retrieve values for the real-time clock (RTC) in the HASP HL Time and HASP HL NetTime keys.

Related Topics

- `hasp_datetime_to_hasptime()`

hasp_login()

Description

Logs into a Feature and thereby establishes a session context.

Syntax

```
hasp_status_t HASP_CALLCONV hasp_login(  
    hasp_feature_t  feature_id,  
    hasp_vendor_code_t vendor_code,  
    hasp_handle_t *  handle  
)
```

Parameters

feature id	Unique identifier for a specific Feature stored in a Sentinel HASP protection key.
vendor code	Pointer to the Vendor Code
handle	Pointer to the session handle

Return Values

HASP_HASP_NOT_FOUND	Required Sentinel HASP protection key not found
HASP_STATUS_OK	Request was successfully completed
HASP_FEATURE_NOT_FOUND	Cannot find requested Feature
HASP_FEATURE_TYPE_NOT_IMPL	Requested Feature type not available
HASP_TMOF	Too many open sessions
HASP_INSUF_MEM	Out of memory
HASP_INV_VCODE	Invalid Vendor Code
HASP_NO_DRIVER	Driver not installed
HASP_NO_VLIB	Vendor library cannot be found
HASP_INV_VLIB	Vendor library cannot be loaded
HASP_OLD_DRIVER	Old driver installed

HASP_UNKNOWN_VCODE	Vendor Code not recognized by the API
HASP_FEATURE_EXPIRED	Feature has expired
HASP_TOO_MANY_USERS	Too many Sentinel HASP protection keys currently connected
HASP_OLD_LM	HASP License Manager version out of date
HASP_DEVICE_ERR	Input/output error in HASP SL secure storage, OR, in the case of a HASP HL key, USB communication error
HASP_TIME_ERR	System time has been tampered with
HASP_TS_DETECTED	Program is running remotely on a Terminal Server
HASP_HARDWARE_MODIFIED	HASP SL key incompatible with machine hardware. HASP SL key locked to different hardware. OR In the case of a V2C file, conflict between HASP SL key data and machine hardware data. HASP SL key locked to different hardware.
HASP_LOCAL_COMM_ERR	Communication error occurred between the application and the local HASP License Manager
HASP_REMOTE_COMM_ERR	Communication error occurred between the local and remote HASP License Managers
HASP_OLD_VLIB	Vendor library too old

Usage Notes

This function establishes a context to a Sentinel HASP protection key containing a license for the requested Feature ID.

When the default Feature ID 0 is used, the API searches only for the Sentinel HASP protection key and ignores the licensing information specified in the key.

The requisite Vendor Codes are stored in a **VendorCodes** folder in your system. Without the correct Vendor Code, the function call cannot succeed. You can open up to 512 simultaneous login sessions.

Related Topics

- `hasp_logout()`
- `hasp_login_scope()`

- Additional information is also available in the *XML Tags* section of the *Sentinel HASP Run-time API Reference* in the Help documentation.

hasp_login_scope()

Description

Logs into a Function to establish a session, according to predefined search parameters.

Syntax

```
hasp_status_t HASP_CALLCONV hasp_login_scope(  
    hasp_feature_t  featureid,  
    char * scope  
    hasp_vendor_code_t  vendor_code,  
    hasp_handle_t *  handle  
)
```

Parameters

feature id	Unique identifier for a specific Feature stored in a Sentinel HASP software protection key.
scope	Definition of the data that is to be searched for the licenses. For more information, refer to the <i>XML Tags</i> section of the <i>Sentinel HASP Run-time API Reference</i> in the Help documentation.
vendor code	Pointer to the Vendor Code
handle	Pointer to the session handle

Return Values

HASP_STATUS_OK	Request was successfully completed
HASP_FEATURE_NOT_FOUND	Requested Feature no longer available
HASP_SCOPE_RESULTS_EMPTY	Unable to locate a Feature matching the scope
HASP_HASP_NOT_FOUND	Sentinel HASP protection key is no longer available
HASP_TMOF	Too many open sessions
HASP_INSUF_MEM	System out of memory
HASP_INV_VCODE	Invalid Vendor Code passed
HASP_NO_DRIVER	Required driver not installed

HASP_OLD_DRIVER	Installed driver too old to execute function
HASP_UNKNOWN_VCODE	Vendor Code not recognized by API
HASP_INVALID_PARAMETER	Scope string too long (max. length 32 KB)
HASP_LOCAL_COMM_ERR	Communication error between application and local HASP License Manager
HASP_REMOTE_COMM_ERR	Communication error between local and remote HASP License Managers
HASP_FEATURE_EXPIRED	Feature has expired
HASP_TOO_MANY_USERS	Too many users currently connected
HASP_OLD_LM	HASP License Manager too old
HASP_DEVICE_ERR	Input/output error in HASP SL secure storage, OR, in the case of a HASP HL key, USB communication error
HASP_TIME_ERR	System time has been tampered with
HASP_FEATURE_TYPE_NOT_IMPL	Requested Feature type not implemented
HASP_INV_SCOPE	XML specification invalid
HASP_NO_VLIB	Vendor library cannot be found
HASP_INV_VLIB	Vendor library cannot be loaded
HASP_OLD_VLIB	Vendor library too old

Usage Notes

This function is used to specify conditions that specify where login information is to be searched for.

The requisite Vendor Codes are stored in a **VendorCodes** folder in your system. Without the correct Vendor Code, the function call cannot succeed. You can open up to 512 simultaneous login sessions.

Related Topics

- `hasp_get_info()`
- `hasp_get_sessioninfo()`
- `hasp_login()`
- `hasp_logout()`

Additional information is also available in the “XML Tags” section of the *Sentinel HASP Run-time API Reference* in the Help documentation.

hasp_logout()

Description

Logs out from a context or session.

Syntax

```
hasp_status_t HASP_CALLCONV hasp_logout(  
    hasp_handle_t  handle  
)
```

Parameters

handle	Handle for the session being terminated
--------	---

Return Values

HASP_STATUS_OK	Request was successfully completed
HASP_INV_HND	Invalid input handle
HASP_LOCAL_COMM_ERR	Communication error occurred between the application and the local HASP License Manager
HASP_REMOTE_COMM_ERR	Communication error occurred between the local and remote HASP License Managers

Usage Notes

Use this function to end a connection to an API object. Once logged out from a session, all memory allocated for the session is released. The connection to the HASP License Manager closes if the logged out connection was the last API session.

Related Topics

- `hasp_login()`
- `hasp_login_scope()`

hasp_read()

Description

Reads the memory of a Sentinel HASP protection key.

Syntax

```
hasp_status_t HASP_CALLCONV hasp_read (  
    hasp_handle_t  handle,  
    hasp_fileid_t  fileid,  
    hasp_size_t    offset,  
    hasp_size_t    length,  
    void *         buffer  
)
```

Parameters

handle	Handle for the session
fileid	Identifier for the file that is to be queried. Possible values are FILEID_RO or FILEID_RW for read only or read/write.
offset	Byte offset for the file
length	Number of bytes to be read from the file
buffer	Pointer to the retrieved data

Return Values

HASP_STATUS_OK	Request was successfully completed
HASP_INV_HND	Invalid input handle
HASP_BROKEN_SESSION	Session has been interrupted
HASP_INV_FILEID	Unrecognized file identifier
HASP_SCHAN_ERR	Communication error in secure channel
HASP_MEM_RANGE	Out of memory

HASP_DEVICE_ERR	Input/output error in HASP SL secure storage, OR, in the case of a HASP HL key, USB communication error
HASP_TIME_ERR	System time has been tampered with
HASP_HASP_NOT_FOUND	Sentinel HASP protection key not found
HASP_LOCAL_COMM_ERR	Communication error occurred between the application and the local HASP License Manager
HASP_REMOTE_COMM_ERR	Communication error occurred between the local and remote HASP License Managers

Usage Notes

Use the `hasp_get_size()` function to determine the size of the file you want to read.

Related Topics

- `hasp_get_size()`
- `hasp_write()`

hasp_update()

Description

Writes an update for a Sentinel HASP license.

Syntax

```
hasp_status_t HASP_CALLCONV hasp_update (
    char *   update_data,
    char **  ack_data
)
```

Parameters

update_data	Pointer to the complete update data
ack_data	Pointer to a buffer to retrieve the acknowledge data

Return Values

HASP_INV_UPDATE_DATA	Required XML tags not found, OR contents in binary data missing or invalid
HASP_INV_UPDATE_OBJ	Binary data does not contain an update
HASP_NO_ACK_SPACE	Acknowledge data requested by the update, however the ack_data input parameter is NULL
HASP_KEYID_NOT_FOUND	Sentinel HASP license to be updated not found
HASP_INV_UPDATE_NOTSUPP	Update not supported by the Sentinel HASP protection key
HASP_UNKNOWN_ALG	Unknown algorithm used in V2C file
HASP_INV_UPDATE_CNTR	Update counter is set incorrectly
HASP_TOO_MANY_KEYS	Too many Sentinel HASP protection keys currently connected
HASP_INV_SIG	Signature verification failed
HASP_LOCAL_COMM_ERR	Communication error occurred between the application and the local HASP License Manager
HASP_DEVICE_ERR	Input/output error in HASP SL secure storage, OR, in the case of a HASP HL key, USB communication error

HASP_TIME_ERR	System time has been tampered with
HASP_UPDATE_TOO_OLD	Trying to install a V2C file with an update counter that is out of sequence with the update counter in the Sentinel HASP protection key. The values of the update counter in the file are lower than those in the Sentinel HASP protection key.
HASP_UPDATE_TOO_NEW	Trying to install a V2C file with an update counter that is out of sequence with the update counter in the Sentinel HASP protection key. The first value in the file is more than 1 greater than the value in the Sentinel HASP protection key.
HASP_HARDWARE_MODIFIED	HASP SL key incompatible with machine hardware. HASP SL key locked to different hardware. OR In the case of a V2C file, conflict between HASP SL key data and machine hardware data. HASP SL key locked to different hardware.
HASP_LOCAL_COMM_ERR	Communication error occurred between the application and the local HASP License Manager

Usage Notes

This function writes update information. Note that the Sentinel HASP protection key must be accessible by the local machine.

The update code contains all necessary data to perform the update on the deployed Sentinel HASP protection key including:

1. Where the updated information is to be written
2. The necessary access data — Vendor Code
3. The actual update information

The function returns an acknowledgement code that is signed/encrypted by the update. The code is evidence that an update has been applied to a license. Memory for the acknowledge data is allocated by the API and must be released using `hasp_free()`.

This function is an extension of the main Sentinel HASP Run-time API and is utilized by the Sentinel HASP Remote Update System utility to update Sentinel HASP protection key licenses.

Related Topics

- `hasp_free()`

hasp_write()

Description

Writes to the memory of a Sentinel HASP protection key.

Syntax

```
hasp_status_t HASP_CALLCONV hasp_write(  
    hasp_handle_t  handle,  
    hasp_fileid_t  fileid,  
    hasp_size_t    offset,  
    hasp_size_t    length,  
    void *         buffer  
)
```

Parameters

handle	Handle for the session
fileid	Identifier for the file to write. Possible value is FILEID_RW for read/write.
offset	Byte offset for the file
length	Number of bytes to be written to the file
buffer	Pointer to the retrieved data

Return Values

HASP_STATUS_OK	Request was successfully completed
HASP_INV_HND	Invalid input handle
HASP_BROKEN_SESSION	Session has been interrupted
HASP_INV_FILEID	Unrecognized file identifier
HASP_SCHAN_ERR	Communications error in secure channel
HASP_MEM_RANGE	Out of memory
HASP_TIME_ERR	System time has been tampered with

HASP_DEVICE_ERR	Input/output error in HASP SL secure storage, OR, in the case of a HASP HL key, USB communication error
HASP_ACCESS_DENIED	Access to Feature is denied
HASP_HASP_NOT_FOUND	Required Sentinel HASP protection key not found
HASP_LOCAL_COMM_ERR	Communication error occurred between the application and the local HASP License Manager

Related Topics

- `hasp_get_size()`
- `hasp_read()`

API Status Codes

This section provides a list of possible return codes related to the operation of the Sentinel HASP Run-time API functions, and a description of their meanings.

No	Status Code	Description
0	HASP_STATUS_OK	Request successfully completed
1	HASP_MEM_RANGE	Request exceeds the Sentinel HASP protection key memory range
3	HASP_INSUF_MEM	System out of memory
4	HASP_TMOF	Too many open sessions
5	HASP_ACCESS_DENIED	Access to Feature denied
6	HASP_INCOMPAT_FEATURE	Legacy decryption function cannot work on the Feature
7	HASP_HASP_NOT_FOUND	Sentinel HASP protection key no longer available
8	HASP_TOO_SHORT	Encrypted/decrypted data length too short to execute function call
9	HASP_INV_HND	Invalid handle passed to function
10	HASP_INV_FILEID	Specified File ID not recognized by API
11	HASP_OLD_DRIVER	Installed driver too old to execute function
12	HASP_NO_TIME	Real-time clock (rtc) not available
13	HASP_SYS_ERROR	Generic error from host system call
14	HASP_NO_DRIVER	Required driver not installed
15	HASP_INV_FORMAT	Unrecognized file format for update
16	HASP_REQ_NOT_SUPP	Unable to execute function in this context
17	HASP_INV_UPDATE_OBJ	Binary data passed to function does not contain an update
18	HASP_KEYID_NOT_FOUND	Sentinel HASP license you requested to update not found
19	HASP_INV_UPDATE_DATA	Required XML tags not found. Contents in binary data are missing or invalid.
20	HASP_INV_UPDATE_NOTSUPP	Update request not supported by Sentinel HASP protection key

No	Status Code	Description
21	HASP_INV_UPDATE_CNTR	Update counter not set correctly
22	HASP_INV_VCODE	Invalid Vendor Code passed
23	HASP_ENC_NOT_SUPP	Sentinel HASP protection key does not support encryption type
24	HASP_INV_TIME	Passed time value outside supported value range
25	HASP_NO_BATTERY_POWER	Real-time clock battery out of power
26	HASP_NO_ACK_SPACE	Acknowledge data requested by the update <code>ack_data</code> parameter is NULL
27	HASP_TS_DETECTED	Program running remotely on a terminal server
28	HASP_FEATURE_TYPE_NOT_IMPL	Requested Feature type not implemented
29	HASP_UNKNOWN_ALG	Unknown algorithm used in V2C file
30	HASP_INV_SIG	Signature verification operation failed
31	HASP_FEATURE_NOT_FOUND	Requested Feature no longer available
32	HASP_NO_LOG	Access log not enabled
33	HASP_LOCAL_COMM_ERROR	Communication error between program and local HASP License Manager
34	HASP_UNKNOWN_VCODE	Vendor Code not recognized by API
35	HASP_INV_SPEC	Invalid XML specification
36	HASP_INV_SCOPE	Invalid XML scope
37	HASP_TOO_MANY_KEYS	Too many Sentinel HASP protection keys currently connected
38	HASP_TOO_MANY_USERS	Too many users currently connected
39	HASP_BROKEN_SESSION	Session has been interrupted
40	HASP_REMOTE_COMM_ERROR	Communication error between local and remote HASP License Managers
41	HASP_FEATURE_EXPIRED	Feature expired
42	HASP_OLD_LM	HASP License Manager version too old
43	HASP_DEVICE_ERR	Input/output error occurred in secure storage area of HASP SL key OR In the case of a HASP HL key, USB communication error occurred

No	Status Code	Description
44	HASP_UPDATE_BLOCKED	Update installation not permitted
45	HASP_TIME_ERR	System time has been tampered with
46	HASP_SCHAN_ERR	Communication error occurred in secure channel
47	HASP_STORAGE_CORRUPT	Corrupt data exists in secure storage area of Sentinel HASP protection key
48	HASP_NO_VLIB	Unable to find Vendor library
49	HASP_INV_LIB	Unable to load Vendor library
50	HASP_SCOPE_RESULTS_EMPTY	Unable to locate any Feature matching scope
52	HASP_HARDWARE_MODIFIED	HASP SL key incompatible with machine hardware. HASP SL key locked to different hardware. OR In the case of a V2C file, conflict between HASP SL key data and machine hardware data. HASP SL key locked to different hardware.
53	HASP_USER_DENIED	Login denied because of user restrictions
54	HASP_UPDATE_TOO_OLD	Trying to install a V2C file with an update counter that is out of sequence with update counter in the Sentinel HASP protection key. Values of update counter in file are lower than those in Sentinel HASP protection key.
55	HASP_UPDATE_TOO_NEW	Trying to install a V2C file with an update counter that is out of sequence with the update counter in the Sentinel HASP protection key. First value in file is more than 1 greater than value in Sentinel HASP protection key
56	HASP_OLD_VLIB	Vendor library too old
400	HASP_NO_API_DYLIB	Unable to locate dynamic library for API
401	HASP_INVALID_API_DYLIB	Dynamic library for API is invalid
500	HASP_INVALID_OBJECT	Object incorrectly initialized
501	HASP_INVALID_PARAMETER	Scope string too long (max. length 32 KB)
502	HASP_ALREADY_LOGGED_IN	Logging in twice to same object
503	HASP_ALREADY_LOGGED_OUT	Logging out twice from same object

No	Status Code	Description
525	HASP_OPERATION_FAILED	Incorrect use of system or platform
698	HASP_NOT_IMPL	Requested Feature type not implemented
699	HASP_INT_ERR	Internal error occurred in API

Blinking the HASP HL Key LED

This section describes how you can implement functionality in your custom application to cause a HASP HL key attached to a user's computer to blink on and off.

Overview

In a situation where multiple HASP HL keys are attached to a computer, it may be convenient to cause the LED in a specific key to blink (flash on and off) so that the user can easily identify the key.

The functionality of causing an HL key to blink exists in the Admin Control Center. This topic describes how you can implement this same functionality in your custom application.

Note:

- The **blink** functionality is implemented using an HTTP request. The functionality is not part of the Run-time API. This information is included here because you will most likely want to implement this functionality in the same application that issues calls to the Run-time API.
 - Basic HASP keys do not have a HASP Key ID, so they cannot be made to blink. (They can be made to blink from Admin Control Center, but this is accomplished using an internal enumeration that is not available from other applications.)
-

How to Blink the HASP HL Key

To cause an HL key to start blinking, issue the following HTTP request:

```
http://127.0.0.1:1947/action.html?blinkon=keyID
```

where *keyID* is the HASP Key ID of the HL Key that you want to start blinking.

To cause an HL key to stop blinking, issue the following HTTP request:

```
http://127.0.0.1:1947/action.html?blinkoff=keyID
```

Example

The following command causes the HL key with HASP Key ID 12345678 to start blinking:

```
http://127.0.0.1:1947/action.html?blinkon=12345678
```

Sentinel HASP Run-time Network Activity

This appendix describes the type of network activity that occurs in the communication between:

- an application (protected using Sentinel HASP) and the local HASP License Manager (referred to as “local communications”).
- the local HASP License Manager and one or more remote HASP License Managers (referred to as “remote communications”).

Details regarding local communications and remote communications are provided on the pages that follow.

This chapter 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 HASP communicates via TCP and UDP on socket 1947. This socket is IANA-registered exclusively for this purpose.

In this appendix:

- [Local Communications](#)
- [Remote Communications](#)

Local Communications



This section describes communication between a protected application and the local HASP License Manager service.

A protected application communicates only with HASP License Manager on the computer where the application is running, regardless of whether the HASP HL or SK Key is located on the same computer or on a remote computer.

Note:

Under Windows, HASP License Manager is a service that is launched automatically by **hasplms.exe**. Under Mac OS and Linux, the HASP License Manager is a process launched automatically by **hasplmd**.

HASP License Manager service opens socket 1947 for listening (both for UDP packets and TCP packets).

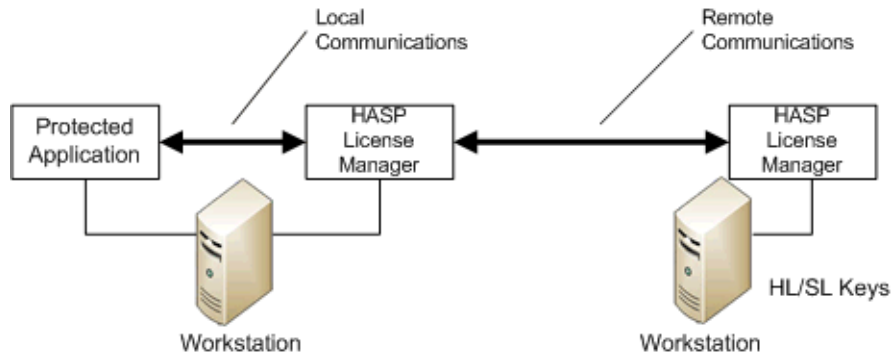
- IPv4 sockets are always opened (HASP License Manager currently does not work without IPv4 installed).
- IPv6 sockets are opened if IPv6 is available.

A protected application tries to connect to 127.0.0.1:1947 TCP to communicate with HASP 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 HASP License Manager service and a remote HASP License Manager service.

This type of communication occurs when the protected application is running on a different computer from the computer where the HASP HL or SL Key is installed.

The protected application communicates only with the local HASP License Manager on the computer where the application is running, as described in [Local Communications on page 334](#). The local HASP License Manager discovers and communicates with the License Manager on the computer containing the HASP Key using one of the following methods:

- The local HASP License Manager issues a UDP broadcast to local subnets on port 1947 using:
 - ◆ IPv4 (always)
 - ◆ IPv6 (if available)

You can disable this broadcast by clearing the **Broadcast Search for Remote Licenses** check box in the Admin Control Center Configuration screen.

- The local License Manager issues a UDP “ping” packet to port 1947 for all addresses specified in the Admin Control Center field **Specify Search Parameters**. These addresses may be individual machine addresses or broadcast addresses.

All 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 HASP 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 License Manager GUID (40 bytes of payload data).

When starting or stopping a License Manager, and when adding or removing a HASP Key, a UDP notification packet is sent, containing the License Manager GUID and a description of the changes encountered. This is done to allow other License Managers to update their data before the next scheduled discovery process.

TCP packets between two License Managers on different computers use HTTP with base-64 encoded data in the body section.

Appendix D

How Sentinel HASP Detects Machine Cloning

This appendix describes the techniques employed by Sentinel HASP to prevent unauthorized use of protected software when the physical or virtual machine on which the software is installed is cloned.

This topic is only relevant for software protected with a HASP SL key. Software that is protected by a HASP HL key is not vulnerable to machine cloning.

For more information on protecting software against cloning, see [Protection Against Cloning on page 149](#).

Overview

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.

When clone detection is enabled for a Product in Sentinel HASP, the Run-time Environment checks for cloning using the criteria described in this appendix. If cloning is detected, Sentinel HASP disables the license. As a result, the end user is unable to operate the software for which a cloned license has been detected.

Note:

This appendix describes the criteria that are built in to Sentinel HASP for clone detection on physical and virtual machines. Your organization may prefer to apply different techniques or different policies for dealing with clone detection. In this case, you have the option of disabling clone detection in Sentinel HASP, and instead, developing your own clone detection and handling routines using the methods available in the Sentinel HASP Business Studio API.

Clone Detection for a Physical Machine

As part of the Activation process for a licensed Product, the Sentinel HASP Run-time 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, including hard drive serial numbers and the motherboard ID. This 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 HASP database.

Each time the end user starts the protected software, the Sentinel HASP Run-time creates a new fingerprint of the computer and compares it to the fingerprint stored in the secure storage.

If the new and stored fingerprints are identical, Sentinel HASP Run-time allows the protected software to operate.

If *either* the hard drive serial number *or* the motherboard ID does not match the characteristics in the fingerprint in the secure storage, Sentinel HASP Run-time still allows the protected software to operate. Sentinel HASP 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 HASP regards computer as a clone and prevents the protected software from operating.(See the table that follows.)

Clone Detection for Physical Machines

		Comparison Results			
Characteristics Compared	Hard drive serial number	Identical	Different	Identical	Different
	Motherboard ID	Identical	Identical	Different	Different
Sentinel HASP Behavior: The software is...		launched	launched	launched	disabled

Clone Detection for a Virtual Machine

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.

Sentinel HASP 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 Sentinel HASP to create a virtual machine fingerprint is absolutely tamper-proof.

Note: The protection against cloning provided by Sentinel HASP 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 HASP Business Studio.

However, when checking the fingerprint for cloning, Sentinel HASP examines all of these characteristics. If *one* (or more) of these characteristics does not match the characteristics in the fingerprint of the license, Sentinel HASP prevents the protected software from operating. Thus, the combination of these parameters in the fingerprint provides protection against cloning. (See the table that follows.)

Clone Detection for Virtual Machines

		Comparison Results			
Characteristics Compared	Virtual MAC Address	Identical	Different	Identical or Different	Identical or Different
	CPU Characteristics	Identical	Identical or Different	Different	Different
	UUID	Identical		Identical or Different	
Sentinel HASP 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 Sentinel HASP to protect against cloning of virtual machines is effective for all types of virtual machine software commonly used by organizations.

How Sentinel HASP Protects Time-based Licenses Locked to HASP SL Keys

This appendix describes the technology used in Sentinel HASP to prevent a user from extending the duration of a software license that is locked to a HASP SL key by adjusting the computer's system clock.

Software that is used in conjunction with HASP SL keys is locked to the machine on which it is activated. The expiration period or date is initially calculated according to the system clock of the machine.

HASP License Manager reads the system time at HASP License Manager startup (by default, part of the machine startup). It subsequently uses its internal running time to calculate the time. When new software that is protected with a HASP SL key is executed for the first time, HASP 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), HASP 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 HASP SL key.
- If the license is to expire on a specific date, HASP License Manager records that date.

Expiration time is calculated in seconds in the format `current HASP License Manager time + number of seconds to expiration`, and the information is stored in the secure storage area of the HASP SL key.

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 the HASP License Manager remains running the changed time will 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 the HASP License Manager is stopped and restarted, for example if the machine is rebooted, it will compare its last recorded internal time with the time of the system clock. When the HASP License Manager detects that the time on the system clock is earlier than that of its internal clock, the HASP SL key is deactivated until such time as the system clock is equal to or later than the time in the License Manager. This means that all applications for which there are licenses on the HASP SL key are blocked until the key is re-enabled.

Re-enabling a Blocked HASP SL Key

To re-enable a blocked HASP SL key, the software vendor sends a new V2C file.

Applying the V2C file to the HASP SL key on the end-user machine resets the flag that indicated time tampering had occurred, making the license available again. In addition, it sets the current time as the reference for future comparison.

Understanding the Sentinel HASP Master Key Licenses

This appendix describes the SafeNet Sentinel HASP model for the Master Key licenses. Its purpose is to assist you in understanding how your Master Key licenses from SafeNet are implemented, and to make decisions about your license update requirements.

The Sentinel HASP Master Key licenses include the following components:

- Product Activation module
- Activations Pool
- Network Seats Pool
- Value of Unlimited Seats
- Reporting module
- Provisional Products module

The components that you purchase depend on your specific requirements and whether you have an in-house Sentinel HASP Business Studio Server installation, or you utilize Sentinel HASP Managed Services.

In this appendix:

- Licensing Concepts
- Activation Module License
- Activations Pool
- Network Seats Pool
- Provisional Products License
- Reporting Module License

Licensing Concepts

In the descriptions of the Master Key licenses model, the following concepts are used:

- **Provisional Product:** A Product that can be used as trialware, or during a grace period. Provisional Products do not require a locking type, since they can be activated and used for a limited period without a Sentinel HASP protection key.
- **Activation:** The process in which a Sentinel HASP Provisional SL key is converted to a locked, computer-specific license. Following activation, the protected application can be used on the end user's computer according to the licenses installed during the activation process.
- **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 HASP License Manager.

For more information about concurrency, see [Specifying the License Terms for Features in a Product](#) on page 152.

Activation Module License

An Activation Module license is required to enable the software activation functionality of Sentinel HASP. The license, together with a pool of activations, provides you with the ability to create a license update file (V2C) that can be applied at the end user site to activate your software on their machine.

Your Activation Module License is either perpetual or issued for a limited time period. This depends on your purchase plan or subscription plan for Sentinel HASP. For more information, consult with your SafeNet sales representative.

You do not require an Activation Module license if you do not intend to create activation files or to enable concurrency for HASP SL keys.

Activations Pool

Each time a Product Key for your software is submitted by an end user, their Provisional license for your software is converted to a locked, machine-specific license, and one activation is consumed. End users can submit a Product Key online, or they can request and receive an activation file to apply manually.

To use this activation functionality, you may need to purchase a pool of activations. (This depends on the nature of your purchase plan or subscription plan for Sentinel HASP.)

When the activation pool is low, you purchase additional activations (if required by your plan). You can configure Sentinel HASP Business Studio to send notifications when the pool reaches a predefined threshold, to ensure that you never run out of activations for your software. For additional information about configuring notifications, refer to the Sentinel HASP Business Studio help.

You do not require activations in the activations pool if you do not intend to create activation files for HASP SL keys.

How Activations and Updates Affect the Activations Pool

The following actions cause Sentinel HASP to deduct activations from your activations pool:

- A provisional SL protection key is activated on an end user's computer.
- An HL or SL protection key is updated (either online or offline) using a Product Key.

The following actions *do not* cause Sentinel HASP to deduct activations from the vendor's activations pool:

- The end user activates an SL protection key more than once using the same Product Key (from an order that was already applied to the key). This would typically occur when end users reformat their hard drive and reinstall the software.
- The end user activates an SL protection key more than once using another Product Key that was created by the same order.
- The Vendor issues a regular HASP Update using a V2C file, without using the Product Key option.
- Activation of a Product Key order that contains only Products that have concurrency enabled. (Only the network seats pool will have seats deducted.)

Additional Information

- Once an activation is consumed, it cannot be returned to the pool. When you purchase activations, SafeNet adds an extra 10% to the number of activations provided, to compensate for situations in which an activation should have been returned to the pool. (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 activations remaining in your activations pool, you will not be able to perform an activation (if your purchase plan or subscription plan requires that you purchase activations).

Network Seats Pool

Network seats are required to enable users to run your software concurrently in a network environment. 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.

To enable concurrency for Features, you may need to purchase network seats for your network seats pool license (if required by your purchase plan or subscription plan). Each time a customer activates your software, the number of concurrent instances that you included in the Product is deducted from the network seats pool. 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 the seats pool is that of the Feature with the highest number of seats.

When the network seats pool is low, you replenish it by purchasing additional network seats (if required by your plan). You can configure Sentinel HASP Business Studio to send notifications when the pool reaches a predefined threshold, to ensure that you never run out of network seats for your software.

You do not require seats in the seats pool if you do not intend to enable concurrency for HASP SL keys.

Note:

Although an activation locks the license to the computer in the network on which HASP License Manager is located, in the event that only “network” licenses (that is, licenses that contain a concurrency value) are included in the order, only the seats pool will be decremented. The activations pool will not be decremented.

How New Activations and Updates of Your Software Affect the Pool

When your protected application is first activated at the customer site, Sentinel HASP 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 network seats pool. (The concurrency in all other Features is ignored.)

For the Sample Product in the graph below, the customer purchased as follows:

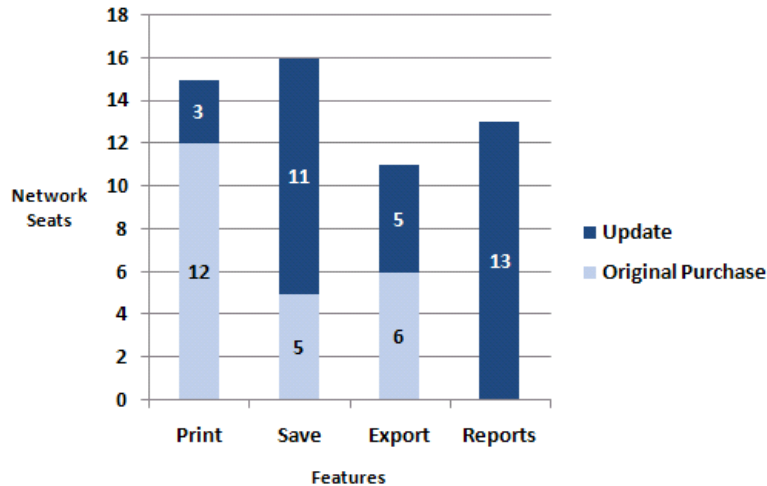
- For the **Print** Feature: 12 network seats
- For the **Save** Feature: 5 network seats
- For the **Export** Feature: 6 network seats

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 fulfil the order, Sentinel HASP calculates the number of seats to deduct from your network seats pool as follows:

1. Sentinel HASP 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 HASP 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 ($16 \text{ total seats} - 12 \text{ already-purchased seats} = 4$).
5. The remainder (4) is the number of seats that is deducted from the network seats pool.

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 HASP calculates the number of seats to deduct from the network seats pool.

Additional Information

- When you purchase seats, SafeNet adds an extra 10% to the number of seats provided, 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 HASP License Manager is located in order to activate on a different computer.
- If you specify the concurrency value for a license as “unlimited” (for example, to create a “site” license), Sentinel HASP deducts from your seat pool the number of seats specified in the **Unlimited Concurrency** license type (also referred to as **Value of Unlimited Seats**) on your Sentinel HASP Master key. This is typically 100 seats.
- If the terms of a license include both an activation and concurrency, both the activations pool and the seats pool are decremented.
- If there are not enough seats in your seats pool, your customer will not be able to perform an activation (if seats are required by your plan).

Provisional Products License

The ability to create and distribute Provisional Products (“trialware”) 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), decide to purchase a license for the application or be blocked from using the application.

Creating Provisional Products is an integral part of the process of protecting software with Sentinel HASP SL protection keys. A vendor who purchases an Activation Module license or who subscribes to Sentinel HASP Managed Services is automatically granted a license to create and distribute Provisional Products for the same period.

A vendor who only distributes software protected by Sentinel HASP HL protection keys can also take advantage of the benefits provided by creating and distributing trialware. However, in this case, the vendor must purchase a Provisional Products license separately from SafeNet in order to create trialware. The Provisional Products license for this type of vendor is typically issued for a specific amount of time.

The ability to create and distribute Provisional Products is also included in the Sentinel HASP Software Protection and Licensing Starter Kit and Developer Kit. Vendors who are experimenting with Sentinel HASP can learn first-hand about Provisional Products.

Reporting Module License

The Reporting facility in Sentinel HASP Business Studio provides a variety of reports that will help vendors increase revenue, improve their customer service, and work more efficiently. The Reporting facility provides software vendors with the ability to produce real-time reports with valuable business information.

Using this feature, managers can 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.

Use of the Reporting facility requires the Reporting Module license. This license is typically issued for a specific amount of time.

The ability to generate and view sample reports is included in the Sentinel HASP Software Protection and Licensing Starter Kit and Developer Kit. Vendors who are experimenting with Sentinel HASP can learn first-hand about the Reporting facility.

For information on the Reporting facility, see Chapter 13, *Generating Sentinel HASP Reports*.

Glossary

Activation counter	Licensing element indicating the number of times a Feature, licensed using Sentinel HASP, can be run
AES	Advanced Encryption Standard (AES) algorithm that is the basis for the Sentinel HASP encryption and decryption
Anti-debugging	Measures applied by the Sentinel HASP system to block potential attacks intended to undermine the protection scheme
API call history	Log of all calls to the Sentinel HASP Run-time API executed using Sentinel HASP ToolBox
API samples	Sample applications that utilize the Sentinel HASP Run-time API. A learning tool used for implementing the Sentinel HASP Run-time API.
Background checks	Random checks executed by protected applications for a required Sentinel HASP protection key
Backward compatibility	Ability to share data or commands with applications protected with earlier HASP versions. Sentinel HASP backward compatibility includes the ability to read and write data, set real-time clocks, and process other 'legacy' commands.
Base Product	An original Product that has been created from scratch from which other Products may be created. All Modification Products, Provisional Products and Cancellation Products are created from Base Products.
Batch Code	Unique character string that represents a Vendor Code. Used in defining Features, Products and orders. It is also used for ordering Sentinel HASP protection keys. With HASP HL keys, the code is printed on the HASP HL key label.
C2V file	Customer-to-Vendor file. A file sent by the customer to the vendor, containing data about deployed Sentinel HASP protection keys.

Cancellation Product	A Product that cancels the licensing details of another 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.
Cross-locking	Indicates that protection can be applied to both HASP HL and HASP SL keys
DataHASP	Utility for protecting data files that are accessed by programs protected by Sentinel HASP Envelope
Decryption	Process of decrypting data that has been encrypted
Default Feature	Feature that is always available in a Sentinel HASP protection key. It requires no configuration.
Demo Vendor Code	See <i>DEMOMA</i>
DEMOMA	Batch Code used for evaluation purposes with any Sentinel HASP application. Its corresponding Vendor Code is available in the VendorCodes folder of your Sentinel HASP installation.
Detach	Temporarily remove a license from a network pool on a host machine for attachment to a remote recipient machine
Distribution Channel	Mechanism that enables a vendor to restrict access for the vendor's distributors to specific Products and orders.
Encryption	Translation of data into a confidential code. To read an encrypted file, you must have the correct encryption engine for decrypting the file.
Encryption engine	Encryption engine in a Sentinel HASP protection key—based on the AES algorithm
Encryption key	Key used for encrypting a data file used with Sentinel HASP Envelope
Encryption level	Number of iterations that the Sentinel HASP Envelope executes with the Sentinel HASP protection key for each interaction
Envelope	See <i>Sentinel HASP Envelope</i>
Envelope template	Defined default protection settings and other project-related data
Expiration date	Date after which a protected program or Feature stops running
Feature	An identifiable functionality of a software application that can be independently controlled by a license. In Sentinel HASP, a Feature may be an entire application, a module or a specific functionality such as Print, Save or Draw.

Feature ID	Unique identifier for a Sentinel HASP-protected Feature
File filter	Defines the files that are exposed to on-the-fly file encryption
Grace period	An initial period of time during which a Product can be used without a Sentinel HASP protection key. See also <i>Provisional Product</i> .
H2R file	Host-to-Recipient file that contains one or more detached Products and their licenses for temporary attachment to a recipient machine
Handle	Unique identifier for accessing the context of a Sentinel HASP login session
HASP HL Basic key	Standard HASP HL local key that is used to protect software, and has a perpetual license. It does not have any memory functionality.
HASP HL Demo key	Sample HASP HL key provided for evaluating Sentinel HASP protection and licensing software. Always has the Batch Code DEMOMA.
HASP HL Drive	A HASP HL key that combines the copy-protection and licensing capabilities of the HASP HL Max key with the convenience of a mass storage drive
HASP HL key	The hardware-based protection and licensing component of Sentinel HASP. One of the Sentinel HASP protection key types.
HASP HL Max key	HASP HL local key with large storage capacity
HASP HL Net key	HASP HL network key
HASP HL NetTime key	HASP HL network key with a real-time clock
HASP HL Pro key	HASP HL local key with moderate storage capacity
HASP HL Time key	HASP HL local key with a real-time clock
HASP ID Number	Unique identity number for a Sentinel HASP protection key
HASP License Manager	Acts as a server and monitors concurrent usage according to the licenses stored in a HASP HL Net key
HASP Memory	Secure memory that resides within a Sentinel HASP HL or SL protection key, for use by the protected software. HASP memory can be accessed or modified using the Run-time 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.

HASP SL key	The software-based protection and licensing component of Sentinel HASP—a virtual HASP HL key
HASP Update	File containing update information for deployed Sentinel HASP protection keys. See also <i>V2C file</i>
Key	See <i>Sentinel HASP protection key</i>
License	Digital permit stored in a Sentinel HASP protection key
License Manager	See <i>HASP License Manager</i>
License terms	Detailed conditions contained in a license
Locking type	Determines the level of protection for a Product, according to the type of Sentinel HASP protection key supplied with the Product
Memory data	Data (for example: passwords, values used by the software) that is specified in memory and transferred to the Sentinel HASP protection key
Modification Product	A modified version of an existing Product
Order	A request for Products or HASP Updates to be shipped to a customer
Product	A licensing entity that represents one of a vendor's marketable software products. The Product is coded into the memory of a HASP key and contains one or more Features. License terms are defined for each Feature in a Product.
Product Key	A string generated by Sentinel HASP Business Studio and supplied to the end user for use as proof of purchase for Product Activation or Update Activation
Production	The implementation of an order for Products or HASP Updates
Protect Once— Deliver Many™	The concept of separation between engineering and business processes, on which Sentinel HASP is designed
Provisional Product	A Product that can be used as trialware, or during a grace period. Provisional Products do not require a locking type, since they can be activated and used for a limited period without a Sentinel HASP protection key.
Real-time Clock (RTC)	Clock available in the HASP HL Time key and HASP HL NetTime key

Recipient machine	Remote machine to which a license that has been detached from a network pool on a host machine is temporarily attached
Reverse Engineering	Software attacks intended to unravel the algorithms and execution flow of a target program by tracing the compiled program to its source code. Sentinel HASP Envelope protection implements contingency measures to repel such attacks and prevent hackers from discovering algorithms used inside protected software.
RUS	See <i>Sentinel HASP Remote Update System</i>
Secure Storage	Area reserved by Sentinel HASP on a computer's local hard drive when one or more Sentinel HASP 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 HASP components.
Sentinel HASP	Software protection and licensing system
Sentinel HASP Admin Control Center	Customizable, Web-based, end-user utility that enables centralized administration of HASP License Managers and Sentinel HASP protection keys
Sentinel HASP Business Studio	Role-based application used to generate licenses and lock them to Sentinel HASP protection keys, write specific data to the memory of a Sentinel HASP protection key, and update licenses already deployed in the field
Sentinel HASP Developer key	A vendor-specific HASP HL key containing the confidential codes assigned by SafeNet. The key is used by the software engineers when protecting programs using Sentinel HASP.
Sentinel HASP Developer Kit	Kit containing software, hardware and documentation for evaluating the Sentinel HASP system
Sentinel HASP Envelope	Application that wraps an application in a protective shield, ensuring that the protected application cannot run unless a specified Sentinel HASP protection key is accessible by the program
Sentinel HASP Master key	A vendor-specific HASP HL key containing the confidential codes assigned by SafeNet. The key is connected to the Sentinel HASP Business Studio Server.
Sentinel HASP protection keys	HASP HL keys and HASP SL keys

Sentinel HASP Remote Update System (RUS)	Enables licenses in deployed Sentinel HASP protection keys to be securely, remotely updated, or the contents of the keys to be modified. See also <i>C2V file</i> and <i>V2C file</i>
Sentinel HASP Run-time API	Interface for inserting calls to a Sentinel HASP protection key
Sentinel HASP Run-time Environment	System component that enables communication between a protected program and a Sentinel HASP protection key
Sentinel HASP ToolBox	Sentinel HASP GUI application designed to facilitate software engineers' use of the Sentinel HASP Run-time API and to generate source code
Sentinel HASP Vendor keys	The Sentinel HASP Master key and Sentinel HASP Developer key that contain your confidential and unique Vendor Codes. These keys enable you to apply protection to your programs, program the Sentinel HASP protection keys that you send to your end users, and to specify the license terms under which your software can be used.
Status code	Error or status message returned by the Sentinel HASP system
Trialware	Software that can be distributed without a Sentinel HASP protection key for end-user evaluation during a limited time period. See also <i>Provisional Product</i> .
UTC	Coordinated Universal Time—the standard time common to every place in the world
V2C file	Vendor-to-Customer file that contains HASP Update data for delivery to end users. This data can include detailed changes to the license terms and/or data to be stored in the end users' Sentinel HASP protection keys.
Vendor Code	A confidential, vendor-unique string containing vendor-specific secrets that enables access to the vendor-specific Sentinel HASP protection keys.

Symbols

- .NET assemblies
 - considerations 86
 - global Feature 86
 - Method-level protection 87
 - method-specific settings 88
 - obfuscation 90
 - protecting 85
 - required RTE libraries 208

A

- about the license model 345
- Activating Products
 - about 120
 - manually 191
 - with HASP HL keys 135, 149
 - with HASP SL keys 135, 149
- activations
 - deducted from the activations pool 348
 - Master Key licenses 347
- Admin Control Center
 - about 217
 - administrator's workflow 220
 - configuration 221
 - interface 219
 - launching 218
- Administration
 - functions 125
 - role 122, 186, 200
 - tasks 186
- AES decryption
 - See* Decryption

- AES encryption
 - See* Encryption

- API libraries
 - generating 61
 - merging 61

- Attacks
 - clock tampering 107
 - cloning hardware keys 106
 - defense against 104
 - emulating protection keys 105
 - modifying key memory 104
 - patching executables 104
 - using terminal servers 106

B

- Base Product 148
- Batch Codes
 - about 39
 - DEMOMA 125, 186
 - for Features 143
 - for orders 165
 - for Products 147
 - for RUS 182, 194
 - introducing in Sentinel HASP 186
 - Sentinel HASP user access 188
- Blink, function to cause HL key to 332
- Branding RUS 182, 194
- Broadcast Search for Remote Licenses
 - check box 335
- Bundles
 - See* Provisional Products
- Business Studio
 - about 118
 - evaluating 125, 186

- roles 121
- window 126

Business Studio Server
See Server

C

C2V

- checking in files 172
- data for cancellation 159
- data in files 172
- data in keys 172
- from cloned machine 172
- generating files 172, 182, 194
- storing data on server 172
- viewing data 172

Canceling licenses 159

Cancellation Products

- about 159
- defining 160
- example 160

Checking in C2V files 172

Clone protection 149

- for physical machines 338
- for virtual machines 339

Cloned machines

- checking in C2V 172

Communication

- between License Managers 335
- between protected application and
local License Manager 334

Comparing protection solutions 32

Concurrent instances

- changing settings 158
- counting 153
- in Master Key licenses 346
- license terms 153
- network environment 153
- per order 154, 168
- See also* Network seats

Connection details 190

Cross-locking 37

Customer Services

- function 165
- role 122, 191

Customers

- authorization to manage 191
- Customer ID 166
- defining 166
- locating 166
- orders 166

Customizing RUS 182, 194

D

Data files

- dfcrypt.exe* encryption 82
- encrypting 82
- handling 81

DataHASP utility

- about 109
- encrypting data files 110
- functionality 112
- launching 112
- prerequisites 111

Decryption

- about 45, 51

Default

- locking type 149, 190
- password 186
- user name 186

DEMOMA

- Batch Code 125, 186

Detachable licenses

- configuring in Features 223
- preparing recipient machine 181

Developer key

- See* Vendor keys

Developer kit 23

Development

- activities 179
- functions 179
- role 122, 165, 179

Distributing

- RUS 182
- See* End user software

Distribution Channel users

- about 121
- restrictions on 123
- See also* Users

- Distribution Channels
 - about 123
 - required authorization to manage 122
- Distribution list for reports 201
- DLL for Run-time Environment
 - installer 181
- Duplicating
 - Products 156

E

- Encryption
 - about 45, 51
 - data files 82
 - DataHASP utility 109
 - dfcrypt.exe* 82
- End user software
 - about 207
 - haspdinst.exe* 214
 - HASPUUserSetup.exe* 214
 - merge modules 210
- Envelope
 - .NET considerations 86
 - authorization to operate 122
 - customizable parameters 76
 - encrypting data files 82
 - functionality 71
 - Java considerations 98
 - Linux prerequisites 93
 - Mac prerequisites 96, 98
 - mandatory parameters 76
 - prerequisites 78
 - protecting .NET assemblies 85
 - protecting Java executables 97, 100
 - protecting Mac binaries 96
 - protecting Win32 programs 80
 - protecting Windows x64 programs 80
 - running from command-line 84, 94, 96
 - using Features 77
 - workflow 73
- Examples
 - defining Features 132
 - defining Products 134
 - license cancellation 160

- license terms 139
- Modification Product 158
- order for HASP HL keys 173
- order for HASP Update 175
- order for Product Keys 174
- order using HASP SL 174
- protection levels 139
- Provisional Products 183
- remote update with RUS 195
- trial use 183

EXE file

- protecting, *See* Envelope
- Run-time Environment installer 181
- with V2C data 180, 195

Export

- Admin Control Center format 182
- CPP-style header 182
- C-style header 182
- CSV format 182
- Feature data 144
- XML format 182

F

Features

- about 118
- Batch Codes 143
- defining 143
- detachable licenses 153
- example 132
- exporting 144
- Feature ID 144
- Feature name 143
- identifying 132
- in Envelope 77
- in Products 133, 147
- license terms 138
- status values 144
- transferring data 144
- withdrawing 145

Files

- C2V 159, 172
- default locations 190
- EXE with V2C data 170, 178, 180
- export 144
- Product Keys 169, 178
- Protecting, *See* Envelope

- Provisional Products 180
- Run-time Environment EXE 181
- Run-time Environment installer DLL 181
- V2C 170, 178, 180
- fingerprint 41, 338
- Fonts, configuring for scheduled reports 201
- Formatting HASP HL keys 173

G

- Grace periods
 - about 137
 - Provisional Products 155

H

- HASP HL keys
 - causing to blink on and off programatically 332
- HASP License Manager
 - See* License Manager
- HASP memory
 - See* Memory
- HASP search mode 77
- HASP Updates
 - implementing 120
- HASP user account 122
- HASP HL keys
 - formatting 173
 - order example 173
 - orders 168, 177
 - Product activation 135, 149
 - protection 135, 149
 - updating licenses with RUS 196
- HASP SL keys
 - order example 174
 - orders 169
 - Product activation 135, 149, 169
 - protection 135, 149
 - Run-time Environment 169, 180
 - tamper protection for Time-based licenses 343
 - updating licenses with RUS 196

HASP Updates

- about 170
- applying remotely 170
- applying with RUS 195
- locating keys 170
- order example 175
- orders 170
- RUS example 195
- haspdinst.exe 214
- hasplmd process 334
- HASPUserSetup.exe 214
- HTTP
 - for TCP packets between two License Managers 336

I

- IANA-registered socket 333
- ID
 - customer 166
 - Feature 144
 - order 166
- Installing
 - Provisional Products 180
 - Run-time Environment 180
 - Server 125
- Introducing Vendor keys 125, 189
- IPv4 sockets 334
- IPv6 sockets 334

J

- Java
 - behavior of protected executables 100
 - protecting executables 97
 - required RTE libraries 208
- Java executables
 - considerations 98

K

- Key
 - See* Protection keys
 - See* Vendor keys

L

License Manager

- communication between local and remote LMs 335

- Run-time network activity 333

License terms

- about 138
- assigning values 153
- canceling 159
- concurrent instances 153
- example 139
- Modification Products 158
- network access 153
- remote desktop users 153
- remote updates 170
- revoking 159
- selecting license type 153
- specifying 153
- transferring 159
- updating 170
- values per order 153, 168
- virtual machines 153

License types

- about 138
- assigning values 153
- selecting 153
- values per order 153

Licensing

- about 130
- fundamentals 33
- of Sentinel HASP 345
- planning 34, 118, 131
- solutions 34

Licensing models

- detailed description of 241
- overview 235
- preparing licensing plan 131

Licensing Plan functions 142

Linux

- distributing Run-time Environment 215

List of reports 203

Locating keys for update 170

Locking Sentinel HASP users 188

Locking types

- about 134

- default 149, 190

- HASP HL only 135, 149

- HASP HL or HASP SL 136, 149

- HASP SL only 135, 149

- Products in orders 167

- selecting 148

M

Mac

- distributing Run-time Environment 214

- protecting binaries 96

MAC address

- clone prevention 340

Master key

- See* Vendor keys

Master Key licenses 345

- activations pool 347

- network seats pool 349

Memory

- about 140, 357

- defining 154

- storing data 154

- utilizing 53

Merge modules 210

Modification Products

- about 157

- defining 158

- example 158

- license terms 158

N

Network

- access to license 153

- concurrent instances 153

Network activity (Run-time) in

- Sentinel HASP 333

Network seats

- pool of, described 349

O

Obfuscation in .NET assemblies 90

Obsolete Product 156

Order Management

- functions 165
- role 122, 165, 176

Orders

- about 119
- customers 166
- defining 166
- examples 173
- for HASP HL keys 168, 177
- for HASP SL keys 169
- for HASP Updates 170
- for Product Keys 169, 178
- holding 171
- in production queue 171
- including Products 167
- Order ID 166
- priority 171
- producing 177
- Product locking types 167
- reference data 171
- saving data 171
- status values 171

P

Passwords (Sentinel HASP)

- assigning to users 188
- changing 124
- default 186
- receiving 124

Product activation 169

- manual 191
- with HASP HL keys 135, 149
- with HASP SL keys 135, 149

Product Keys

- about 169
- authorization to manage 191
- files 169, 178
- order example 174
- orders 169, 178
- Product activation 191
- proof of purchase 169
- server verification 169
- use with HASP SL 169

Product Management role 122, 141

Production

- functions 164

orders for HASP HL keys 177

orders for Product Keys 178

orders in queue 171

role 122, 165, 176

Products

- about 119
- activation 120
- association with Distribution Channel 123
- Base Product 148
- Batch Codes 147
- Cancellation Products 159
- defining 147
- duplicating 156
- example 134
- grace, *See* Provisional Products
- in orders 167
- including Features 133, 147
- locking types 135, 148
- memory 154
- Modification Products 157
- Obsolete Products 156
- Product name 147
- Provisional Products 155
- reference data 147
- restoring 156
- status values 155
- trial, *See* Provisional Products
- types 148
- withdrawing 156

Protect Once–Deliver Many 36

Protection

- against cloning 149
- against copying 30
- API, *See* Run-time API
- attack types 104
- DataHASP utility 109
- defense against attacks 104
- elements 50
- encryption and decryption 45, 108
- fundamentals 30, 102
- inserting multiple calls 107
- intellectual property 30
- Java executables 97
- Mac binaries 96
- Method-level 87
- options 54

- programs and data files 50
 - selecting method 53
 - solutions 31
 - solutions, combined 33
 - solutions, comparison 32
 - using checksum verification 108
 - using Run-time API and Envelope 107
- Protection keys
 - attributes 42
 - for orders 167
 - HASP HL keys 41
 - HASP SL keys 41
 - locating for update 170
 - locking 134
 - memory 154
 - searching for 77
 - selecting 39
 - updating 170
- Protection levels
 - example 139
 - HASP HL keys 135, 149
 - HASP SL keys 135, 149
 - per order 136, 149
- Provisional Products
 - defining 155
 - description 137
 - example 183
 - generating bundles 180
 - generating V2C file 181
 - in orders 167
 - installing 169, 180
 - key ID 180
 - license terms 155, 180
 - license to create 352
 - output files 180
 - properties 155
 - Vendor library 180
- R**
- Read/write HASP memory 140, 154
- Read-only HASP memory 140, 154
- Recycling HASP HL keys 173
- Remote desktop
 - access to license 153
- Remote License Manager 335
- Remote Update System
 - See* RUS
- Report Generation role 200
- Reports
 - about the Reports facility 200
 - configuring font for scheduled reports 201
 - export formats 202
 - granting permissions to work with 200
 - license to generate 353
 - list of available 203
 - presentation formats 201
 - scheduling 201
- Restoring obsolete Products 156
- Roles
 - Administration 122, 186, 200
 - Customer Services 122, 191
 - Development 122, 165, 179
 - for Sentinel HASP users 188
 - Order Management 122, 165
 - Product Management 122
 - Production 122, 165
 - Report Generation 200
- RTE
 - See* Run-time Environment
- Run-time API
 - about 57
 - functionality 68
 - login function 66
 - planning requirements 64
 - prerequisites 59
 - samples 63
 - ToolBox 62
 - workflow 65
- Run-time Environment
 - command-line installer 181
 - distributing for Linux 215
 - distributing for Mac 214
 - distributing to end users 209
 - EXE file 181
 - generating installer 180–181
 - initializing 169
 - installer API 181
 - installer DLL 181
 - installer PKG 181
 - Mac PKG 181

- required libraries for .NET and Java 208

- updating 210

- Run-time network activity 333

- RUS

- applying HASP Updates 195

- Batch Code 182, 194

- branding 182, 194

- customizing 182, 194

- description 182, 193

- distributing 182, 194

- example 195

- executable 194

- generating C2V files 182, 194

- instructions for end users 196

- processing V2C files 170, 178, 195

- with HASP HL keys 196

- with HASP SL keys 196

S

- Seats (network) 349

- Secure storage

- about 359

- See Network seats

- Sentinel HASP Developer key

- See Vendor keys

- Sentinel HASP Business Studio

- See Business Studio

- Sentinel HASP Envelope

- See Envelope

- Sentinel HASP Master key

- See Vendor keys

- Sentinel HASP protection keys

- See Protection keys

- Sentinel HASP Remote Update System

- See RUS

- Sentinel HASP Run-time Environment

- See Run-time Environment 209

- Sentinel HASP Run-time API

- See Run-time API

- Sentinel HASP ToolBox

- See ToolBox

- Sentinel HASP users

- See Users (Sentinel HASP)

- Sentinel HASP Vendor keys

- See Vendor keys

- Server

- connecting Master key 125

- installing 125

- storing C2V data 172

- verifying Product Keys 169

- socket 1947 333

- Solutions

- combined protection 33

- customizing 38

- protection 31

- protection, comparison 32

- Specify Search Parameters field 336

- Standard users

- accessing Distribution Channel

- entities 124

- described 121

- See also Users

- Starter kit 24

- Status values

- Features 144

- orders 171

- Products 155

- Superuser 122

- Support

- technical 26

- training 26

- System settings 190

T

- Technical specifications 42

- ToolBox

- about 62

- encrypting data 64

- Toolbox

- authorization to operate 122

- Trialware

- about 137

- example 183

- Provisional Products 155

U

- UDP notification packet 336

- UDP packets 336

Unlimited Concurrency license type 352

Updates

See HASP Updates

Updating deployed keys 170

User roles

See Roles

Users (Sentinel HASP)

 access to batches 188

 default user name 186

 defining 187

 Distribution Channel 121

 locking 188

 passwords 124, 188

 preventing access 188

 roles 188

 Standard 121

 user names 124, 187

V

V2C

 data in EXE 170

 default file location 178

 file for Provisional Products 180

 generating files 170, 178, 195

 input to Run-time Environment 181

 launching files 182

 processing with RUS 170, 178

Value of Unlimited Seats license

 component 352

Vendor Code

 about 39

 extracting 60

 MasterHASP Wizard 60

Vendor keys

 Developer key 40

 extracting Vendor Code 60

 introducing 125, 189

 maintaining Master keys 188

 Master key 40

Vendor library 180

Vendor Suite 124

Vendor-to-Customer file

See V2C

Virtual MAC address

 clone protection 340

Virtual machines

 access to license 153

 clone protection 339

W

Win32 programs

 behavior of protected programs 80

 data file protection 81

Windows x64 programs

 data file protection 81

Windows x64 programs

 behavior of protected programs 80

Withdrawing

 Features 145

 Products 156

