

www.aladdin.ru

HASP<sup>®</sup> HL

Кроссплатформенная аппаратно-программная система для защиты программ и данных

# Руководство разработчика

 Aladdin<sup>®</sup>  
SECURING THE GLOBAL VILLAGE  
eAladdin.com

---

## **АВТОРСКИЕ ПРАВА И ТОРГОВЫЕ МАРКИ**

Система HASP® и её документация защищены авторскими правами© с 1985 по настоящее время компанией Aladdin Knowledge Systems Ltd. HASP®, MacHASP® и MemoHASP® являются зарегистрированными торговыми марками Aladdin Knowledge Systems Ltd.

Все права защищены.

NetHASP, TimeHASP™, HASP36™, MemoHASP36™, NetHASP36™, USBHASP™ и AladdinCARD™ являются торговыми марками Aladdin Knowledge Systems Ltd.

Все другие торговые марки, марки и названия изделий, используемые в данном руководстве, являются торговыми марками соответствующих владельцев.

## **ОГРАНИЧЕННАЯ ГАРАНТИЯ, ОГРАНИЧЕНИЕ ОТВЕТСТВЕННОСТИ И ВОЗМЕЩЕНИЯ**

Гарантия Aladdin Knowledge Systems Ltd. (Aladdin) распространяется на программное обеспечение и устройства HASP на период двенадцати (12) месяцев со дня покупки в порядке, указанном в Лицензионном соглашении с Разработчиком.

Гарантийные обязательства и ограничения ответственности Aladdin указаны в Лицензионном соглашении.

**ЗА ИСКЛЮЧЕНИЕМ УКАЗАННОГО ВЫШЕ, НЕ СУЩЕСТВУЕТ ИНЫХ ЯВНЫХ ИЛИ ПОДРАЗУМЕВАЕМЫХ ГАРАНТИЙ НА ПРОДУКТЫ ALADDIN, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ ТАКОВОЙ, ПОДРАЗУМЕВАЕМУЮ ГАРАНТИЮ СПРОСА И ПРИГОДНОСТИ ДЛЯ ОПРЕДЕЛЁННОЙ ЦЕЛИ.**

Были приложены все усилия, чтобы информация в данном документе была полной и точной. Aladdin не несёт ответственности за прямые или косвенные убытки или ущерб в результате неточностей или упущений.

Спецификации в данном документе могут быть изменены без уведомления.

## **ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ МЕЖДУ РАЗРАБОТЧИКОМ И ALADDIN KNOWLEDGE SYSTEMS LTD.**

**ВАЖНАЯ ИНФОРМАЦИЯ – ПОЖАЛУЙСТА, ВНИМАТЕЛЬНО ПРОЧТИТЕ ДАННОЕ СОГЛАШЕНИЕ ПРЕЖДЕ, ЧЕМ ОТКРЫТЬ ПАКЕТ И/ИЛИ ИСПОЛЬЗОВАТЬ ЕГО СОДЕРЖИМОЕ И/ИЛИ ПРЕЖДЕ, ЧЕМ ЗАГРУЖАТЬ ИЛИ УСТАНОВЛИВАТЬ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ.**

---

ВСЕ УКАЗАНИЯ ПО ИСПОЛЬЗОВАНИЮ ПРОДУКТОВ HASP (включая без ограничений Комплект Разработчика, библиотеки, утилиты, дискеты, CD-ROM, устройства HASP® и Руководство Разработчика) (далее «Продукт»), ПРЕДОСТАВЛЯЕМЫЕ КОМПАНИЕЙ ALADDIN KNOWLEDGE SYSTEMS LTD. (или любым дочерним предприятием — каждое из них упоминаемое как «ALADDIN»), ПОДЧИНЯЮТСЯ И БУДУТ ПОДЧИНЯТЬСЯ УСЛОВИЯМ, ОГОВОРЕННЫМ В ДАННОМ СОГЛАШЕНИИ.

ОТКРЫВАЯ ПАКЕТ, СОДЕРЖАЩИЙ ПРОДУКТ, И/ИЛИ ЗАГРУЖАЯ ДАННОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ (как определено далее по тексту) И/ИЛИ УСТАНОВЛИВАЯ ДАННОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ НА ВАШ КОМПЬЮТЕР И/ИЛИ ИСПОЛЬЗУЯ ДАННЫЙ ПРОДУКТ, ВЫ ПРИНИМАЕТЕ ДАННОЕ СОГЛАШЕНИЕ И СОГЛАШАЕТЕСЬ С ЕГО УСЛОВИЯМИ.

ЕСЛИ ВЫ НЕ СОГЛАСНЫ С ДАННЫМ СОГЛАШЕНИЕМ, НЕ ОТКРЫВАЙТЕ ЭТОТ ПАКЕТ И/ИЛИ НЕ ЗАГРУЖАЙТЕ И/ИЛИ НЕ УСТАНОВЛИВАЙТЕ ДАННОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И НЕМЕДЛЕННО (по меньшей мере, в течение 7 дней с даты получения этого пакета) ВЕРНИТЕ ЭТОТ ПРОДУКТ В ALADDIN, СОТРИТЕ ДАННОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ВСЕ ЕГО ЧАСТИ СО СВОЕГО КОМПЬЮТЕРА И НЕ ИСПОЛЬЗУЙТЕ ЕГО НИКОИМ ОБРАЗОМ. ПРИ ВОЗВРАЩЕНИИ ПРОДУКТА С КОПИЕЙ ПЛАТЕЖНОГО ДОКУМЕНТА ВАМ БУДЕТ ВОЗВРАЩЕНА ЕГО СТОИМОСТЬ.

## 1. Права и Собственность

ДАННОЕ ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ НЕ ЯВЛЯЕТСЯ СОГЛАШЕНИЕМ О ПРОДАЖЕ. Программное обеспечение пакета Aladdin's HASP Product Development Kit, включая все переработки, исправления, модификации, дополнения, обновления и/или усовершенствования к ней (далее по всему тексту и любой его части определяемое как «Программное обеспечение»), и связанная с ним документация НЕ ПРЕДНАЗНАЧЕНЫ ДЛЯ ПРОДАЖИ и являются и останутся безраздельной собственностью Aladdin. Все права на интеллектуальную собственность и на Продукт (включая, без ограничений, авторские права, профессиональные секреты, торговые марки и т.д.), подтвержденные или включенные в приложенные/ взаимосвязанные/ имеющие отношение к данному руководству, данные, содержащиеся в нём, являются и будут являться собственностью исключительно компании Aladdin. Данное Соглашение не передает Вам права на Программное обеспечение, а лишь предоставляет ограниченное право на использование, которое подлежит отмене согласно условиям данного Лицензионного соглашения. Ничто в данном Соглашении не подтверждает отказ Aladdin от прав на интеллектуальную собственность по какому бы то ни было закону.

---

## 2. Лицензия

По выплате взносов за права, Aladdin настоящим предоставляет Вам, а Вы принимаете личную, неэксклюзивную и полностью могущую быть отозванной ограниченную Лицензию на использование данного Программного обеспечения только в исполнительской форме, как описано в прилагаемой к Программному обеспечению документации и только в соответствии с условиями данного Соглашения:

- i.** Вы можете установить Программное обеспечение и использовать его на компьютерах, расположенных на месте Вашего бизнеса, как описано в соответствующей документации от Aladdin;
- ii.** Вы можете путём слияния присоединить Программное обеспечение к программам Вашего компьютера с единственной целью, описанной в данном Руководстве программиста; однако любая часть Программного обеспечения, объединённая с другой компьютерной программой, неизбежно будет считаться его производной и будет по-прежнему соотноситься с условиями настоящего Соглашения.
- iii.** Вы можете создать определенное количество копий Программного обеспечения — не более трех (3) — исключительно в целях резервного копирования и для нужд разработки. Программное обеспечение не должно использоваться в любых иных целях.

## 3. Передача лицензии

В случае объединения данного Программного обеспечения с иными компьютерными программами согласно разделу 2 Вы можете передать разрешение в соответствии с условиями данного Соглашения на использование подключенного Программного обеспечения и перепродать аппаратное обеспечение, состоящее из устройств HASP<sup>®</sup>, которые Вы приобрели у Aladdin, дистрибьюторам и/или пользователям.

Предваряя подобную продажу и передачу лицензии, Вы обязаны включить особую ссылку в Ваш контракт с таковыми дистрибьюторами и/или пользователями, а также во всём остальном обеспечить подчинение дистрибьюторов и/или пользователей условиям гарантии и лицензии, оговоренным в данном Соглашении компанией Aladdin.

## 4. Запрещённое использование

За исключением указанного в разделах 1, 2 и 3, Вы соглашаетесь:

- i.** не использовать, не модифицировать Программное обеспечение, не передавать разрешение на данное Программное обеспечение и

---

любые другие Продукты Aladdin, за исключением явно указанного в данном Соглашении и Руководстве программиста;

- ii. не продавать, не передавать лицензию, не сдавать в аренду, не передавать, не переводить, не закладывать, не разделять Ваши права в рамках данного Разрешения с кем/ кому-либо ещё;
- iii. не модифицировать, не демонтировать, не декомпилировать, не реконструировать, не видоизменять и не расширять данное Программное обеспечение и не пытаться раскрыть исходную систему кодирования данного Программного обеспечения;
- iv. не помещать данное Программное обеспечение на сервер с возможностью доступа к нему через открытую сеть;
- v. не использовать какие бы то ни было резервные или архивные копии данного Программного обеспечения (или позволять кому-либо ещё использовать такие копии) с любой иной целью, кроме замены оригинальной копии, в случае его разрушения или дефектности.

Если Вы являетесь членом Европейского Союза, данное соглашение не затрагивает Ваших прав согласно любому законодательству, осуществляющему Директиву Совета ЕС о Правовой защите Компьютерных Программ. Если Вы стремитесь получить какую бы то ни было информацию в рамках значения данной Директивы, прежде всего Вам следует обратиться в Aladdin.

## 5. Ограниченная гарантия

Aladdin гарантирует, что:

- i. данное Программное обеспечение с момента доставки его Вам в течение трёх (3) месяцев будет работать в достаточном соответствии с Руководством разработчика, при условии, что оно будет использоваться на компьютерном аппаратном обеспечении и с операционной системой, для которой оно было разработано;
- ii. что устройство HASP® в течение двенадцати (12) месяцев с даты поставки будет в достаточной мере избавлено от значительных дефектов в материалах и конструктивных характеристиках.

## 6. Отказ от гарантии

ALADDIN НЕ ГАРАНТИРУЕТ, ЧТО ЛЮБОЕ ИЗ ЕГО ИЗДЕЛИЙ БУДЕТ СООТВЕТСТВОВАТЬ ВАШИМ ТРЕБОВАНИЯМ, ИЛИ ЧТО ЕГО РАБОТА БУДЕТ БЕСПЕРЕБОЙНОЙ ИЛИ БЕЗОШИБОЧНОЙ. В РАМКАХ ЗАКОНА ALADDIN ОТКРЫТО ОТКАЗЫВАЕТСЯ ОТ

---

ВСЕХ ГАРАНТИЙ, НЕ ОГОВОРЕННЫХ ЗДЕСЬ И ВСЕХ ПОДРАЗУМЕВАЕМЫХ ГАРАНТИЙ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ ТАКОВОЙ, ПОДРАЗУМЕВАЕМУЮ ГАРАНТИЮ СПРОСА И ПРИГОДНОСТИ ДЛЯ ОПРЕДЕЛЁННОЙ ЦЕЛИ. НИ ОДИН ИЗ ДИЛЕРОВ, ДИСТРИБЬЮТЕРОВ, ПРОДАВЦОВ, АГЕНТОВ ИЛИ СОТРУДНИКОВ ALADDIN НЕ УПОЛНОМОЧЕН ПРОИЗВОДИТЬ МОДИФИКАЦИИ, РАСШИРЕНИЯ ИЛИ ДОПОЛНЕНИЯ К ДАННОЙ ГАРАНТИИ. Если Вы произвели какие-либо модификации Программного обеспечения или любой из частей данного Продукта во время гарантийного периода; если средства и устройство HASP® подвергаются механическим повреждениям, аварии, неосторожному или неправильному обращению; или если Вы нарушаете любое из условий настоящего Соглашения, то гарантия, упомянутая выше в разделе 5, будет немедленно прекращена. Гарантия не действительна, если Программное обеспечение используется на или в сочетании с аппаратным обеспечением или программой иной, чем немодифицированная версия аппаратного обеспечения или программы, в сочетании с которыми по своей конструкции должно было использоваться данное Программное обеспечение, как описано в данном Руководстве разработчика.

## 7. Ограничение возмещения

В случае нарушения гарантии, оговоренной выше, единственным обязательством Aladdin будет, руководствуясь лишь усмотрением Aladdin:

- i.** заменить или бесплатно отремонтировать Продукт или его составляющие, если это не противоречит вышеупомянутому ограничению гарантии;
- ii.** возместить стоимость, выплаченную Вами за Продукт или его составляющие. Любая замененная или отремонтированная составляющая будет на гарантии или в течение оставшегося времени от начального гарантийного периода, или в течение 30 дней, если срок начального гарантийного периода истекает ранее. Гарантийные требования должны быть выставлены в письменном виде в течение гарантийного периода, но не позднее семи (7) дней с момента обнаружения дефекта, и содержать в себе подтверждения, удовлетворяющие Aladdin. Вся Продукция должна быть возвращена дистрибьютору, через которого была совершена покупка (если покупка состоялась не непосредственно в Aladdin), и отправлена возвращающей стороной с оплаченной стоимостью перевозки и страховки. Продукция или составляющие таковой должны быть отправлены с копией Вашего чека.

---

## 8. Исключение косвенных убытков

Стороны признают, что Продукт по сути своей сложный и не может быть полностью лишен ошибок. ALADDIN НЕ НЕСЁТ ОТВЕТСТВЕННОСТИ (КАК ПО ДОГОВОРУ, ДЕЛИКТУ, ВКЛЮЧАЯ ХАЛАТНОСТЬ, ТАК И В ЛЮБОЙ ИНОЙ ФОРМЕ) ПЕРЕД ВАМИ ИЛИ ЛЮБОЙ ТРЕТЬЕЙ СТОРОНОЙ ЗА ЛЮБЫЕ ПОТЕРИ ИЛИ УБЫТКИ (ВКЛЮЧАЯ КОСВЕННЫЕ, ЭКСТРЕННЫЕ ИЛИ ПОБОЧНЫЕ УБЫТКИ), ВКЛЮЧАЯ, БЕЗ ОГРАНИЧЕНИЙ, ЛЮБЫЕ ПОТЕРИ ИЛИ УБЫТКИ ПРИБЫЛЬНОСТИ БИЗНЕСА, ПОТЕРЮ ДОХОДНОСТИ ИЛИ ПРЕСТИЖА, ПОТЕРЯННУЮ ИЛИ ПОВРЕЖДЁННУЮ ИНФОРМАЦИЮ ИЛИ ДОКУМЕНТАЦИЮ ВСЛЕДСТВИЕ КАКОГО-ЛИБО ИСПОЛЬЗОВАНИЯ ДАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И/ИЛИ ЛЮБОЙ СОСТАВЛЯЮЩЕЙ ДАННОЙ ПРОДУКЦИИ, ДАЖЕ ЕСЛИ ALADDIN ОСВЕДОМЛЁН О ВОЗМОЖНОСТИ ПОДОБНЫХ УБЫТКОВ.

## 9. Ограничение ответственности

В СЛУЧАЕ, ЕСЛИ, НЕСМОТРЯ НА УСЛОВИЯ ДАННОГО СОГЛАШЕНИЯ, ALADDIN ПРИЗНАН ОТВЕТСТВЕННЫМ ЗА УБЫТКИ НА ОСНОВАНИИ КАКИХ-ЛИБО ДЕФЕКТОВ ИЛИ НЕСООТВЕТСТВИЯ ЕГО ПРОДУКЦИИ, ПОЛНАЯ ОТВЕТСТВЕННОСТЬ ЗА КАЖДУЮ ДЕФЕКТНУЮ ЕДИНИЦУ ПРОДУКЦИИ НЕ БУДЕТ ПРЕВЫШАТЬ СУММУ, ВЫПЛАЧЕННУЮ КОМПАНИИ ALADDIN ЗА ЭТУ ДЕФЕКТНУЮ ЕДИНИЦУ ПРОДУКЦИИ.

## 10. Никаких иных гарантий

За исключением вышесказанного, в соответствии с преамбулой, Aladdin не дает никаких ни явных, ни подразумеваемых гарантий на качество и производительность продукта, а также на его соответствие при использовании тем или иным способом.

## 11. Прекращение действия

В случае невыполнения Вами условий данного Соглашения действие Вашей Лицензии и настоящего Соглашения будет прекращено. По прекращении действия данного Лицензионного Соглашения, произведённому компанией Aladdin:

- i. разрешение, предоставленное Вам в данном Соглашении, будет недействительным, и Вы, по прекращении срока действия, не сможете продолжать дальнейшее использование данного Программного обеспечения и других лицензионных Продуктов;

- 
- ii. Вы незамедлительно вернёте компании Aladdin всё материальное имущество, представляющее право Aladdin на интеллектуальную собственность и все копии такового и/или сотрёте/удалите любую информацию, содержащуюся в них в электронном виде. Разделы 1, 4, 6-12 будут продолжать действовать даже в случае прекращения действия настоящего Соглашения.

## **12. Действующий Закон и Законодательство**

Данное Соглашение должно быть истолковано и определено в соответствии с законами Израиля (за исключением конфликта применения правовых норм), и только израильский суд уполномочен отправлять правосудие в любых конфликтах и спорах, возникающих из данного Соглашения. Применение Конвенции Объединённых Наций о Договорах Международной Продажи Товаров (the United Nations Convention of Contracts for the International Sale of Goods) однозначно исключается. Неспособность любой из сторон привести в исполнение любое из прав, предоставленных по данному тексту, или принять меры против другой стороны в случае любого нарушения, обозначенного по тексту, не должно рассматриваться как отказ этой стороны от последующего приведения в исполнение прав или совершения последующих действий в случае дальнейших нарушений.

## **13. Программное обеспечение третьих сторон**

Если Продукт содержит в себе любое программное обеспечение, предоставленное какой-либо третьей стороной, такое программное обеспечение от третьей стороны предоставляется «как оно есть» без какой-либо гарантии, и разделы 2, 3, 6, 8, 9-13 настоящего Соглашения применяются ко всем поставщикам программного обеспечения от таких третьих сторон и к программному обеспечению от третьих сторон, как если бы они были Aladdin и Продукт соответственно.

## **14. Разное**

Настоящее Соглашение представляет собой полное соглашение, касающееся Разрешения, и может быть изменено только посредством письменного соглашения, данного обеими сторонами. Если выполнение какого-либо условия настоящего Соглашения представляется невозможным, такое условие будет скорректировано только в пределах, обеспечивающих возможность выполнения данного условия.

Демонстрационные ключи, входящие в Комплект Разработчика, в стоимость комплекта не включаются и должны быть возвращены в Aladdin (либо в любое представительство Aladdin) при покупке первой партии рабочих ключей.

Я ПРОЧЁЛ И ПОНЯЛ НАСТОЯЩЕЕ СОГЛАШЕНИЕ О РАЗРЕШЕНИИ И СОГЛАСЕН С ИСПОЛНЕНИЕМ ВСЕХ ЕГО УСЛОВИЙ.

---

## Соответствие CE



Все продукты HASP соответствуют Директиве CE EMC и соответствующим стандартам\*. Продукты HASP маркируются логотипом CE, а CE карта соответствия HASP включается в каждый комплект поставки или предоставляется по запросу.

\* EMC директива 89/336/ЕЕС и соответствующие стандарты EN 55022, EN 50082-1.

## Соответствие FCC

Уполномоченные органы FCC определили, что HASP не является Периферийным вычислительным устройством класса «В» и, таким образом, не требует соответствия нормам FCC.

## Соответствие UL

Все продукты HASP успешно прошли тесты UL 94 на воспламеняемость пластических материалов для частей устройств и приборов. Продукты HASP соответствуют нормам безопасности оборудования информационных технологий UL 1950 (UL 1950 Safety of Information Technology Equipment regulations).

## Соответствие ISO9002



Все продукты HASP разработаны и выпускаются компанией Aladdin Knowledge Systems, сертифицированной по стандарту ISO 9002.

Система обеспечения качества Aladdin одобрена Международной организацией по стандартизации (ISO), что является подтверждением того, что стандарты выпускаемых продуктов и оказываемых компанией Aladdin услуг постоянно соответствуют спецификациям, предназначенным для обеспечения полного удовлетворения клиентов.

## Сертификат соответствия

По запросу Aladdin Knowledge Systems предоставит Сертификат соответствия любому разработчику программного обеспечения, который желает продемонстрировать, что все продукты HASP соответствуют установленным спецификациям. Разработчики программного обеспечения могут предоставлять этот сертификат конечным пользователям вместе со своими программами.

# Содержание

<b>Об этом руководстве</b> .....	xvii
<b>Глава 1. Введение</b> .....	1
<b>Преимущества системы HASP HL</b> .....	2
Существенная гибкость .....	2
Автоматическое лицензирование .....	3
Преимущества для ваших клиентов .....	3
Сервис и техническая поддержка по всему миру .....	4
<b>Электронные ключи HASP HL</b> .....	5
Модели ключей HASP HL .....	5
Демонстрационный Комплект HASP HL .....	6
Мастер Комплект HASP HL .....	7
<b>Глава 2. Ключевые концепции HASP HL</b> .....	9
<b>Ваши уникальные коды и ключи HASP HL</b> .....	9
Автоматическая система лицензирования .....	12
<b>Как работает защита HASP HL?</b> .....	13
<b>Методы защиты с помощью HASP HL</b> .....	15
<b>Лицензирование с помощью HASP HL</b> .....	16
Возможности лицензирования .....	17
<b>Порядок работы с HASP HL</b> .....	18
<b>Глава 3. Установка HASP HL</b> .....	21
Существующее программное обеспечение HASP HL .....	21
<b>Краткое описание защиты программного обеспечения с помощью HASP HL</b> .....	23
<b>Установка HASP HL в среде Windows</b> .....	28
Структура устанавливаемых компонентов .....	29
Установка драйверов устройств HASP HL .....	29

---

<b>Установка HASP HL в операционной системе Mac</b> .....	32
<b>Установка HASP HL в операционной системе Linux</b> .....	36
<b>Получение Кодов Разработчика</b> .....	39
<b>Глава 4. Защита программного обеспечения</b> .....	41
<b>Защита с помощью HASP HL</b> .....	42
Методы защиты с помощью HASP HL .....	43
Шифрование с помощью алгоритма AES .....	43
Использование памяти ключа HASP HL .....	44
Антиотладочные и антиреинжиниринговые средства .....	45
<b>Выбор метода защиты</b> .....	45
<b>Глава 5. Защита с помощью системы HASP HL API</b> .....	47
Универсальный API .....	48
<b>Условия, необходимые для применения API</b> .....	48
<b>Обучение применению HASP HL API</b> .....	50
HASP HL ToolBox .....	50
Примеры API .....	51
<b>Реализация защиты с помощью HASP HL API</b> .....	51
Функция Login в HASP HL API .....	54
<b>Доступные функции HASP HL API</b> .....	57
<b>Глава 6. Защита с помощью HASP HL Envelope</b> .....	59
<b>Возможности HASP HL Envelope</b> .....	59
<b>Условия применения HASP HL Envelope</b> .....	61
<b>Работа с HASP HL Envelope</b> .....	61
Основная процедура защиты .....	61
Работа с HASP HL Envelope в режиме командной строки .....	62
Параметры защиты с помощью HASP HL Envelope .....	64
<b>HASP HL Envelope для приложений Mac</b> .....	67
<b>Шифрование файлов данных</b> .....	68

---

<b>Глава 7. Стратегии защиты</b> .....	71
<b>Защита с помощью HASP HL – наилучшие подходы</b> .....	72
<b>Оптимизация защиты с помощью HASP HL API</b> .....	73
Аспекты защиты программного обеспечения .....	73
Оптимальная реализация защиты с помощью HASP HL API .....	74
Работа с кодом приложения и HASP HL API .....	77
<b>Наилучшие способы применения HASP HL Envelope</b> .....	79
<b>Глава 8. Обзор лицензирования с помощью HASP HL</b> .....	81
<b>Ключевые концепции лицензирования с помощью HASP HL</b> .....	81
<b>Доступные средства лицензирования HASP HL</b> .....	84
Аппаратные средства .....	84
Программное обеспечение .....	84
<b>Планирование процесса лицензирования с помощью HASP HL</b> .....	85
Функции лицензирования .....	87
<b>Лицензирование с помощью HASP HL в действии</b> .....	88
<b>Глава 9. Лицензирование с помощью HASP HL Factory</b> .....	91
<b>Условия применения HASP HL Factory</b> .....	92
<b>Структуры данных HASP HL Factory</b> .....	94
Серии ключей .....	94
Функции .....	94
Пакеты .....	95
Команды .....	97
<b>Создание команд HASP HL</b> .....	99
<b>Выполнение команд</b> .....	102
Выполнение команды с помощью Системы дистанционного перепрограммирования .....	103
<b>Просмотр ключей HASP HL</b> .....	103
Считывание времени с ключей HASP HL .....	104

---

<b>Глава 10. Система дистанционного перепрограммирования (RUS – Remote Update System)</b> .....	105
<b>Концепция</b> .....	105
<b>Компоненты</b> .....	106
<b>Порядок работы с HASP HL RUS</b> .....	108
<b>Утилита HASP HL RUS</b> .....	109
Применение HASP HL RUS .....	109
Маркировка V2C–файла .....	111
<b>Глава 11. Передача HASP HL вместе с вашим программным обеспечением</b> .....	113
<b>Программное обеспечение HASP HL, предназначенное для конечных пользователей</b> .....	113
Передача драйверов HASP HL под Windows .....	114
Windows Update .....	115
Применение Aladdin DiagnostiX .....	116
Модули слияния .....	116
API для установки драйвера HASP HL .....	118
haspdinst.exe .....	119
HASPUserSetup.exe .....	119
Передача пользователям демонов HASP HL для MAC .....	121
Передача конечным пользователям демона HASP HL для Linux .....	124
<b>Глава 12. HASP License Manager</b> .....	127
<b>Обзор возможностей HASP License Manager</b> .....	127
<b>HASP License Manager для Windows</b> .....	128
Установка HASP License Manager в операционной системе Windows .....	128
<b>Запуск и прекращение работы HASP License Manager</b> .....	129
<b>Работа с HASP License Manager</b> .....	130
<b>HASP License Manager для Mac</b> .....	132
<b>HASP License Manager для Linux</b> .....	134
<b>Установка HASP License Manager</b> .....	135

---

---

<b>Настройка HASP License Manager</b> .....	136
Параметры настройки HASP License Manager .....	137
<b>Настройка клиента HASP HL Net</b> .....	143
<b>Разделы файла конфигурации</b> .....	144
Определение ключевых слов .....	144
<b>Настройка среды HASP HL Net</b> .....	148
Настройка времени ожидания .....	150
<b>Определение числа обслуживаемых защищенных приложений</b> .....	151
<b>Глава 13. Aladdin Monitor</b> .....	153
<b>Установка Aladdin Monitor</b> .....	154
Настройка Aladdin Monitor .....	154
Мониторинг работы HASP License Manager .....	154
<b>Проверка ключей HASP HL</b> .....	155
<b>Служба HASP License Manager</b> .....	158
<b>Глава 14. Диагностика ключей HASP HL</b> .....	159
Функции диагностики утилиты Aladdin DiagnostiX .....	159
Диагностика ключей HASP HL .....	160
Создание отчетов о ключах HASP HL .....	163
Интеграция с внешним инструментом генерирования отчетов .....	164
Обновление драйверов HASP HL .....	165
<b>Aladdin DiagnostiX Memory Beamer</b> .....	166
<b>Глава 15. Устранение неисправностей</b> .....	169
<b>Список контрольных вопросов</b> .....	170
<b>Проблемы и решения</b> .....	171
<b>Глава 16. Глоссарий по HASP HL</b> .....	173
<b>Глава 17. Справочник по HASP HL API</b> .....	181
<b>hasp_datetime_to_hasptime()</b> .....	184
<b>hasp_decrypt()</b> .....	185

---

<b>hasp_encrypt()</b> .....	186
<b>hasp_free()</b> .....	187
<b>hasp_get_rtc()</b> .....	188
<b>hasp_get_sessioninfo()</b> .....	189
<b>hasp_get_size()</b> .....	190
<b>hasp_hasptime_to_datetime()</b> .....	191
<b>hasp_legacy_decrypt()</b> .....	192
<b>hasp_legacy_encrypt()</b> .....	193
<b>hasp_legacy_set_idletime()</b> .....	194
<b>hasp_legacy_set_rtc()</b> .....	195
<b>hasp_login()</b> .....	196
<b>hasp_logout()</b> .....	198
<b>hasp_read()</b> .....	199
<b>hasp_update()</b> .....	201
<b>hasp_write()</b> .....	203
<b>Коды статуса API</b> .....	205
<b>Глава 18. Спецификации ключей HASP HL</b> .....	207

# Список таблиц

Модели ключей HASP HL	5
Пример перечня функций	19
Пример перечня пакетов	19
Команды утилиты Haspdinst	31
Параметры командной строки для aksusbd (Mac)	35
Ключи командной строки для aksusbd (Linux)	38
HASP HL API в сравнении с HASP HL Envelope	46
Параметры командной строки для HASP HL Envelope	63
Установки конфигурационного файла HASP HL Envelope	67
Перечень возможных команд dfcrypt.exe	69
Лицензионные возможности аппаратных средств HASP HL	84
Пример перечня функций	86
Примеры пакетов и их содержание	87
Доступные программные номера различных моделей ключей HASP HL	95
Исходные файлы инсталляционного пакета для установки демона в среде Mac OSX	123
Ключи HASP License Manager	136
Последовательность поиска nhsrv.ini	137
Логические значения для файла nhsrv.ini	138
Информация о HASP License Manager	155
Информация о ключе HASP	155
Информация о ключе HASP HL	156
Таблица программ	156
Таблица открытых сессий	157
Вкладка Key Access History утилиты Aladdin DiagnostiX	162
Перечень функций HASP HL API	182
Технические спецификации ключей HASP HL	207
Технические характеристики моделей HASP HL	208



# Об этом руководстве

*Руководство по защите Программного Обеспечения и Лицензированию с Помощью HASP HL* предназначено для оказания помощи разработчикам в защите и лицензировании их программного обеспечения с использованием системы HASP HL. В руководстве содержится информация о том, как система HASP HL может наилучшим образом удовлетворить потребности в защите и лицензировании вашего программного обеспечения.

Данное руководство разделено на четыре части.

**Часть I – Введение в систему HASP HL.** В данном разделе рассматривается система HASP HL, ее основные концепции лицензирования и защиты программного обеспечения, процесс установки системы. С информацией раздела следует ознакомиться сразу после того, как вы откроете Демонстрационный Комплект или Мастер Комплект HASP HL.

**Часть II – Защита с помощью HASP HL.** Раздел представляет собой обзор и подробное описание методов защиты HASP HL, в нем также рассматриваются стратегии повышения степени защиты вашего программного обеспечения с помощью системы HASP HL. Данный раздел предназначен для разработчиков программного обеспечения, заинтересованных в использовании обоих методов защиты HASP HL.

**Часть III – Лицензирование с помощью HASP HL.** В разделе содержатся предложения по эффективному использованию программного обеспечения HASP HL в целях лицензирования защищаемого программного обеспечения. Данный раздел будет особенно интересен менеджерам по продукту компаний, занимающихся изданием программного обеспечения. Кроме того, раздел может быть полезен непосредственным исполнителям и иным лицам, занятым в производстве программных продуктов. Если вы планируете лицензировать ваше программное обеспечение, мы настоятельно рекомендуем ознакомиться с этой частью, чтобы понять, как именно HASP HL может использоваться наилучшим образом в данном случае.

**Часть IV – Распространение HASP HL.** Раздел содержит информацию о том, как программное обеспечение HASP HL может доставляться конечным пользователям для обеспечения оптимальности совместной работы защищаемого программного обеспечения и аппаратных средств HASP HL. В разделе также рассматриваются различные способы и средства эффективного распространения имеющихся программных компонентов HASP HL.

В данном руководстве важные комментарии, предложения и предупреждения сопровождаются следующими символами:

Символ	Значение
	Предупреждение
	Важный комментарий или совет
	Полезная идея или предложение, позволяющие повысить производительность системы HASP HL.

Для удобства применения данного руководства в его конце приводится глоссарий, объясняющий основные понятия системы HASP HL.

Также в руководстве содержатся спецификации, информация по API системы HASP HL и руководство по устранению неисправностей.

# Введение

Добро пожаловать в систему HASP HL!

В данной главе рассматриваются следующие темы:

- введение в систему обеспечения защиты и лицензирования с помощью HASP HL, характеристики этой системы и ее преимущества;
- представление линейки аппаратных и программных средств HASP HL;
- описание содержимого Демонстрационного Комплекта и Мастер Комплекта HASP HL.

## О системе HASP HL

HASP HL является простой в использовании и очень надежной системой, построенной на аппаратных средствах, и предназначенной для лицензирования и защиты программного обеспечения. Система HASP HL предотвращает неавторизованное использование программного обеспечения, защищает интеллектуальную собственность и авторские права, а также предоставляет возможность использования различных моделей лицензирования. Система предоставляет в ваше (разработчика программного обеспечения) распоряжение полный контроль над использованием ПО и поддерживает современные методы продаж, что позволяет увеличить денежные поступления.

Защищенное HASP HL приложение посылает запросы к ключу HASP HL, подключенному к компьютеру. Если возвращаемый HASP HL ответ имеет надлежащую форму, а лицензия, хранящаяся на ключе HASP HL, действительна, приложение продолжает работу. Если ответ неверен, приложение не будет загружено, переключится в демонстрационный режим или некоторые его функции будут недоступными.

Реализация безопасности и лицензирования с помощью HASP HL достаточно проста, однако обеспечиваемый уровень безопасности чрезвычайно высок. Если ваше приложение находится под защитой HASP HL, его запуск будет возможен только в случае подключения к компьютеру соответствующего ключа HASP HL, поставляемого вместе с программным обеспечением.

## Преимущества системы HASP HL

### Существенная гибкость

Система HASP HL предлагает широкий спектр продуктов, решений и функций, позволяющих осуществлять защиту программного обеспечения, включая решения для различных аппаратных платформ. Реализуемая система защиты предусматривает использование ключей с памятью, ключей, предназначенных для работы в сетевой среде, и ключей с функцией отсчета времени.

Лицензирование с помощью HASP HL осуществляется независимо от защиты. В систему HASP HL включены инструменты, которые позволяют изменять условия лицензирования защищаемого программного обеспечения.

### Максимальная простота применения

HASP HL Envelope (автоматическая защита) является быстрым и простым способом реализации защиты программного обеспечения с помощью HASP HL. Использование данного способа защиты подразумевает применение дружественного графического интерфейса.

Краткосрочность необходимого обучения и применение стандартного API во всех продуктах гарантирует простое и быстрое внедрение программных компонентов HASP HL в защищенное приложение. Применение утилиты HASP HL ToolBox дает возможность интерактивного использования и изучения API системы HASP HL. Сам API не зависит от используемой модели HASP HL или аппаратной платформы, на которой планируется использовать приложение. Это в значительной степени упрощает процесс защиты.

### Поддержка широкого спектра программных сред

В комплекте с системой HASP HL поставляются интерфейсы и примеры для многочисленных компиляторов и программных языков, что снижает трудозатраты и ускоряет реализацию защиты с помощью HASP HL.

### Поддержка широкого спектра операционных систем

Системой поддерживаются следующие операционные системы: Windows 98 SE/ME/2000/XP/Server 2003, Mac OS X и Linux.

### Кросс-платформенное решение

HASP HL является решением, независимым от платформы. Один ключ HASP HL может использоваться для защиты приложений Windows, Mac и Linux, что позволяет серьезно сократить время на разработку интегрированной защиты и сократить издержки, связанные с доставкой и логистикой.

---

## Высокий уровень безопасности

### Аппаратная поддержка шифрования с помощью алгоритма AES

В качестве основы операций шифрования и дешифрования в системе HASP HL используется алгоритм AES. Возможности шифрования/дешифрования, реализованные в аппаратных средствах HASP HL, позволяют достичь тесной интеграции аппаратного обеспечения и защищаемых приложений. Функции системы позволяют ставить выполнение критических функций приложения в зависимость от наличия соответствующего ключа.

При использовании в процессе шифрования такого общезвестного алгоритма как AES в основе обеспечения безопасности лежит ключ шифрования. Наши аппаратные средства обеспечивают то, что этот секретный 128-битный ключ шифрования никогда не покидает ключа HASP HL. Каждому покупателю HASP HL присваивается уникальный ключ (или ключи) шифрования.

### Передовые защитные алгоритмы и антиотладочная технология

Программное обеспечение HASP HL использует надежные алгоритмы защиты кода, а также самую современную в отрасли антиотладочную технологию. Реализованные в HASP HL специальные анти-хакерские функции создают практически непреодолимые препятствия для потенциальных взломщиков.

### Зашифрованный обмен данными

В целях повышения безопасности весь обмен данными между приложением и HASP HL случайным образом шифруется, что предотвращает любые попытки эмуляции аппаратного ключа HASP HL.

## Автоматическое лицензирование

HASP HL позволяет разрабатывать и реализовывать различные схемы лицензирования. Определять и заносить лицензии на ключи HASP HL можно с помощью одной утилиты — HASP HL Factory. Необходимость включать условия лицензии в код самого приложения отсутствует. Проверка перечня модулей, на которые распространяется лицензия, времени действия лицензии, количества запусков приложения, числа пользователей и т.д. автоматически осуществляется системой HASP HL и не требует вмешательства разработчика.

## Преимущества для ваших клиентов

Защита, обеспечиваемая HASP HL, выгодна как вам, так и вашим клиентам. Ниже приводятся несколько факторов, обуславливающих выгоду защиты программного обеспечения для ваших клиентов.

### **Окупаемость затрат**

Поскольку защита с помощью HASP HL увеличивает продажи и выручку, вы можете вложить больший объем финансовых ресурсов в дальнейшую разработку и техническую поддержку своего продукта. В свою очередь ваши клиенты получают более современный продукт и техническую поддержку более высокого уровня.

### **Защита лицензионного соглашения**

Защита программного обеспечения позволяет обеспечить соблюдение лицензионного соглашения. HASP HL является наиболее деликатным способом обеспечения соблюдения его условий. Система гарантирует, что клиенту не придется следить за собственными служащими или рисковать нарушением лицензионного соглашения.

### **Защита инвестиций законных пользователей**

HASP HL защищает законных пользователей от тех, кто несанкционированно использует программное обеспечение и не платит за его использование, а также использует при этом ресурсы технической поддержки.

## **Сервис и техническая поддержка по всему миру**

### **Несколько производственных площадок**

Производственные площадки, расположенные на четырех континентах, позволяют обеспечить быструю доставку точно в срок, а так же при необходимости заменить продукты.

### **Локальная техническая поддержка и сервис более чем в 40 странах**

Работа 9 международных офисов и 40 локальных дистрибуторов позволяет предоставить техническую поддержку практически в любое время и в любом месте.

### **Консультационная служба Aladdin**

Для получения более подробных рекомендаций по реализации защиты с помощью HASP HL вы можете связаться с нашей командой международных консультантов. В зависимости от ваших потребностей, они могут провести обучение или оказать помощь по следующим вопросам:

- интеграция HASP HL и вашего продукта;
- анализ и выработка наилучшей стратегии защиты ваших приложений;
- реализация вашей схемы защиты и лицензирования.

## Электронные ключи HASP HL

Ключи HASP HL подключаются к компьютеру через порт USB и имеют несколько разновидностей (см. Таблицу 1.1).

**Таблица 1.1 Модели ключей HASP HL**

Модель ключа HASP HL	Размер памяти	Количество лицензий	Основные характеристики
HASP HL Basic	Нет	1	Шифрование/дешифрование
HASP HL Pro	112 байт	16	Шифрование/дешифрование HASP HL ID
HASP HL Max	4096 байт	112	Шифрование/дешифрование HASP HL ID
HASP HL Time	4096 байт	112*	Шифрование/дешифрование HASP HL ID Часы реального времени
HASP HL Net 10	4096 байт	112	Шифрование/дешифрование HASP HL ID Сетевой доступ
HASP HL Net 50	4096 байт	112	Шифрование/дешифрование HASP HL ID Сетевой доступ
HASP HL Net 250	4096 байт	112	Шифрование/дешифрование HASP HL ID Сетевой доступ
*8 зарезервировано за датой окончания лицензии			

Все ключи HASP HL являются кросс-платформенными USB-устройствами, которые можно использовать для защиты приложений, работающих в среде Windows, Mac и Linux.

### Модели ключей HASP HL

#### **HASP HL Basic – низкие издержки, высокая безопасность**

HASP HL Basic является самым экономичным из предлагаемых нами устройств для обеспечения защиты программного обеспечения. Использование всех свойств системы HASP HL, направленных на обеспечение

максимальной защищенности, сочетается в этом ключе с невысокой ценой.

### **HASP HL Pro и HASP HL Max – ключи, обеспечивающие максимальный уровень защиты**

Две данные модели ключей объединяет одинаково высокий уровень защиты на основе шифрования, а также гибкость, получаемая за счет 4 КВ защищенной памяти, открытой на чтение и запись, а также уникального идентификатора (ID-номера) для каждого ключа. Ключи HASP HL Pro и HASP HL Max могут быть перепрограммированы непосредственно на том месте, где они установлены, что позволяет реализовывать маркетинговую стратегию, основанную на таких способах продаж, как подписка, распространение демо-версий. С помощью одного ключа можно защитить до 112 приложений.

### **HASP HL Time – защита программного обеспечения с использованием часов реального времени**

У HASP HL Time есть внутренние часы, показывающие точное время (часы, минуты и секунды) и дату (день, месяц, год). Ключ был разработан специально для сдачи программного обеспечения в аренду. Он также позволяет периодически выставлять счет клиенту за использование программного обеспечения и его обслуживание. HASP HL Time оснащен 4 КВ защищенной памяти, доступной на чтение и запись и имеет уникальный ID-номер. Ключ HASP HL Time поддерживает до 104 лицензий – 8 лицензий зарезервированы за датами окончания действия лицензий.

### **HASP HL Net – предоставление лицензий в сетевой среде**

Данный ключ является наилучшим решением для различных сетевых сред. HASP HL Net, подключенный к любой сетевой станции, защищает приложение и контролирует количество рабочих станций, использующих его в единицу времени. HASP HL Net имеет уникальный ID-номер, 4 КВ памяти и может хранить до 112 лицензий.

Существует три типа ключей HASP HL Net:

- HASP HL Net 10 – поддерживает работу до 10 пользователей;
- HASP HL Net 50 – поддерживает работу до 50 пользователей;
- HASP HL Net 250 – поддерживает работу до 250 пользователей.

## **Демонстрационный Комплект HASP HL**

Демонстрационный Комплект (ДК) HASP HL содержит программное и аппаратное обеспечение, позволяющее оценить возможности защиты и лицензирования с помощью HASP HL.

---

## Программное обеспечение

Программное обеспечение HASP HL находится на одном компакт-диске.

## Аппаратное обеспечение

В состав Демонстрационного Комплекта, в соответствии с вашим заказом, включены демонстрационные электронные ключи HASP HL Demo.



Демонстрационный ключ не входит в стоимость Комплекта и его следует использовать только в целях оценки возможностей HASP HL. При заказе рабочих ключей HASP HL, вы возвращаете демонстрационные ключи и вам присваивается уникальный Код Разработчика. Для оценки программного обеспечения HASP HL используйте демонстрационные ключи серии DEMOMA.

## Мастер Комплект HASP HL

Мастер Комплект (МК) HASP HL аналогичен Демонстрационному Комплекту. Основное отличие заключается в аппаратном обеспечении. Вашей компании предоставляется уникальное аппаратное обеспечение, которое может использоваться только вместе с вашим Кодом Разработчика. Код содержится на мастер-ключе HASP HL, поставляемом в составе комплекта.

Ключи HASP HL, поставляемые в составе МК, могут применяться для защиты и лицензирования программного обеспечения и могут передаваться конечным пользователям. Можно защитить свое программное обеспечение, а затем просто заказать необходимое количество дополнительных электронных ключей.



# Ключевые концепции HASP HL

В главе представлены основные концепции, лежащие в основе системы HASP HL. В главе рассматриваются следующие темы:

- Ваши уникальные коды и ключи HASP HL.
- Что делает систему HASP HL уникальной?
- Как работает система HASP HL?
- Защита и лицензирование с помощью HASP HL.
- Пример последовательности шагов при работе с HASP HL.

В целях обеспечения эффективности работы системы HASP HL мы рекомендуем вам ознакомиться с концепциями и терминами, описанными в данной главе.

## Ваши уникальные коды и ключи HASP HL

На ключах HASP HL, которые вы заказали в компании Aladdin, содержится уникальная информация, характерная только для вашей компании. Эта информация используется системой HASP HL для обмена данными с вашими ключами, а также для того, чтобы отличать ваши ключи от ключей других разработчиков.

Поставляемые Aladdin специфичные для разных разработчиков ключи и коды описаны ниже.

- **HASP HL Demo:** ключ HASP HL, поставляемый в составе Демонстрационного Комплекта. Используется исключительно для оценки возможностей системы. Все демонстрационные ключи принадлежат Серии DEMOMA. Соответствующий Код Разработчика DEMOMA и иные идентификаторы интегрированы во все инструменты HASP HL. При оценке возможностей системы

HASP HL с помощью демонстрационного ключа HASP HL Demo в мастер-ключе Master HASP HL нет необходимости. При получении первой партии ключей HASP HL с уникальным Кодом Разработчика, демонстрационные ключи должны быть возвращены.

**НЕ РАСПРОСТРАНЯЙТЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, ЗАЩИЩЕННОЕ С ПОМОЩЬЮ КЛЮЧА HASP HL DEMO!**

- **Мастер-ключ Master HASP HL:** особый ключ HASP HL, содержащий уникальные коды и идентификаторы, используемые системой HASP HL, которые присваиваются вам компанией Aladdin. Вы не должны передавать этот ключ кому-либо. Он необходим для извлечения Кода Разработчика, используемого утилитой Vendor Center для защиты и лицензирования программного обеспечения.
- **Серия ключа:** от пяти до восьми символов, являющихся вашим уникальным Кодом Разработчика. Он печатается на наклейке каждого ключа HASP HL, а также на мастер-ключе, который связывается с серией ваших ключей. Серию ключа необходимо указывать при оформлении заказа на ключи у Aladdin. При заказе новой Серии ключа вы также получаете и новый мастер-ключ HASP HL.
- **Код Разработчика:** уникальный код, присваиваемый Aladdin, который извлекается из памяти мастер-ключа HASP HL. В коде содержится информация, необходимая программному обеспечению HASP HL для обмена данными с вашими ключами. Извлекать Код Разработчика с мастер-ключа помогает утилита Vendor Center.

Описанные выше ключи обычно поставляются в качестве части ДК или МК HASP HL. Более подробная информация приводится в разделе «Демонстрационный Комплект HASP HL».

На рисунке 2.1 показана взаимосвязь между мастер-ключами HASP HL и самой системой HASP HL.

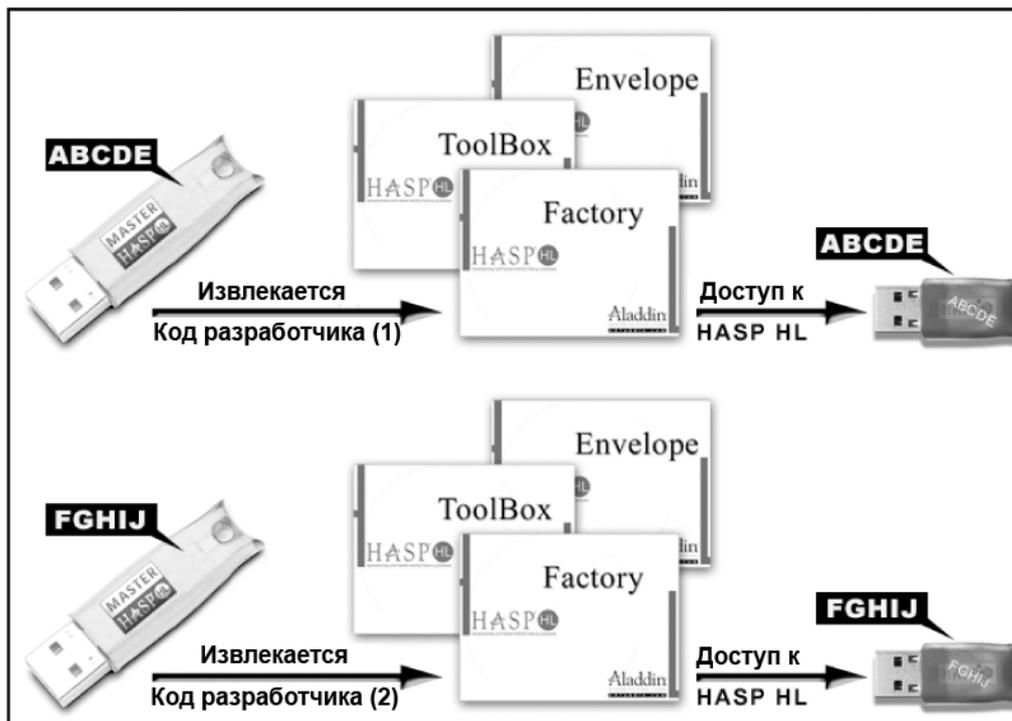
**Рисунок 2.1** Мастер-ключи HASP HL и ключи Разработчика А и В

При размещении первого заказа на ключи вам присваивается уникальный Код Разработчика, который заносится в мастер-ключ HASP HL. На рисунке 2.1 Разработчику А присвоена серия «ABCDE» и предоставлен мастер-ключ соответствующей серии. Все ключи HASP HL, поставляемые данному разработчику, будут маркироваться и программироваться данной серией.

В тоже время Разработчику В была присвоена другая Серия — «FGHIJ». Именно таким набором символов будут маркироваться в будущем все ключи HASP HL этого разработчика. С первой партией ключей HASP HL компания Aladdin всегда посылает соответствующий мастер-ключ.

Каким образом Код Разработчика извлекается с мастер-ключа HASP HL? На рисунке 2.2 Код Разработчика 1, присвоенный Разработчику программного обеспечения А, извлекается с мастер-ключа Серии «ABCDE». Для извлечения Кода Разработчика Разработчик А может воспользоваться любым инструментом Vendor Center. Код Разработчика 1 используется системой HASP HL для получения доступа ко всем ключам Серии «ABCDE». В тоже время Код Разработчика 2, присвоенный разработчику программного обеспечения В, извлекается с мастер-ключа Серии «FGHIJ». Код Разработчика 2 используется системой HASP HL для получения доступа ко всем ключам Серии «FGHIJ».

Рисунок 2.2 Извлечение Кодов Разработчика А и В



## В чем уникальность системы HASP HL?

### Автоматическая система лицензирования

Концепция Автоматической системы лицензирования является основой защиты и лицензирования с помощью HASP HL. Концепция подразумевает полное разделение процесса защиты программного обеспечения и определения моделей продаж, лицензирования.

### Защита

Процесс построения защиты включает в себя обеспечение безопасности приложения и реализацию стратегии безопасности, которая определяется вашими разработчиками. Система HASP HL позволяет делать это единожды. При применении HASP HL менеджеры по продукту совершенно не зависят от команды разработчиков. В случае возникновения потребности в новой модели предоставления лицензий, последняя может быть реализована без участия разработчиков.

Разделение защиты и лицензирования отражается в системе HASP HL в дифференциации инструментов для выполнения этих двух процессов. Разработчикам приходится лишь единожды защитить приложение, а вопросами лицензирования им вообще не приходится заниматься. Они устанавливают, необходимо ли системе HASP HL осуществлять автоматическое лицензирование, но реально процесс лицензирования разработчиками не проводится (осуществляется системой HASP HL автоматически). Как следствие, процесс построения защиты ускоряется и становится более интуитивным.

Менеджеры по продукту не попадают в зависимость от разработчиков каждый раз, когда им необходимо реализовать новую модель продаж. Им просто следует применить определенные инструменты HASP HL, которые позволяют определить порядок лицензирования приложения. По заданным менеджерами шаблонам генерируются лицензии и обновления к ним, которые затем используются с Системой Дистанционного Перепрограммирования. Такая схема обеспечивает большую гибкость и свободу в разработке и реализации моделей продаж, а также позволяет быстрее реагировать на изменяющиеся требования рынка.

В графическом виде разделение защиты и лицензирования в системе HASP HL показано на рисунке 2.3.

## Как работает защита HASP HL?

### Приложения, связанные с аппаратным обеспечением HASP HL

В основе системы HASP HL лежит тесная интеграция защищаемого приложения и самой системы.

Защита строится на предоставлении доступа к защищаемому приложению только в случае присутствия соответствующего ключа HASP HL.

С точки зрения запуска приложения, защита HASP HL подразумевает выполнение простой процедуры:

1. защищенное приложение посылает запрос на подключенный к компьютеру ключ HASP HL;
2. если возвращаемый ключом ответ соответствует ожидаемому, приложение запускается;
3. если возвращаемый ответ неверен, приложение либо не загружается, либо переключается в демонстрационный режим, либо ограничиваются его функции.

### Идентификация ключа HASP HL

Ключи HASP HL содержат информацию, которая является специфичной именно для вашей компании, что обеспечивает уникальность ключей,

распространяемых вместе с защищенным вами программным обеспечением.

Наличие соответствующего ключа может быть проверено следующими способами:

- с использованием аппаратного крипто-процессора;
- с помощью проверки специфического идентификатора ключа;
- с использованием функций памяти ключа.

## **Использование крипто–процессора**

Защита с помощью HASP HL строится на основе проверок наличия соответствующего ключа. Система HASP HL осуществляет такие проверки, выполняя шифрование и дешифрование данных с помощью механизма шифрования, использующего алгоритм AES.

Проверка наличия ключа HASP HL с использованием функций шифрования и дешифрования требует планирования. Данные, которые во время работы посылаются на ключ для дешифрования, требуют предварительного шифрования.

Дешифрованные данные могут быть проверены с использованием данных защищаемого приложения. Более подробная информация по этому вопросу приведена в разделе «Шифрование и дешифрование данных с помощью ключа HASP HL». После дешифрования данные используются защищенным приложением. Зашифрованные данные являются функцией от данных, посланных на ключ HASP HL и Кода Разработчика. Таким образом, шифрование одной и той же строки с помощью двух различных кодов дает разные результаты.

Для шифрования данных можно использовать либо API системы HASP HL, либо HASP HL ToolBox. Более подробная информация по этому вопросу представлена в разделах «Утилита HASP HL ToolBox» и «Реализация API системы HASP HL».

## **Использование возможностей памяти ключей HASP HL**

Все ключи HASP HL, за исключением HASP HL Basic, имеют защищенную внутреннюю память, доступную на чтение/запись. Память ключей может использоваться в следующих целях:

- для контроля доступа к различным модулям программного обеспечения или различным программным пакетам;
- для присвоения уникального кода каждому пользователю программного обеспечения;
- для хранения лицензий;
- для сохранения паролей, программного кода, переменных и прочих данных.

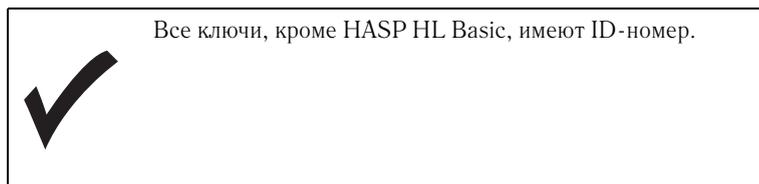
Информация о наличии и размере памяти различных моделей ключей HASP HL приведена в таблице 1.1.

Редактирование памяти ключа может осуществляться либо с помощью HASP HL ToolBox, либо с помощью HASP HL Factory. Для получения более подробной информации следует обратиться к справочной системе данных утилит.

### Проверка ID-номера ключа HASP

Каждый ключ HASP HL с памятью имеет уникальный ID-номер (идентификатор). Защищенное приложение может осуществлять его проверки.

Идентификаторы позволяют различать пользователей вашего приложения. По результатам проверки приложением ID-номера можно предугадать различную реакцию на наличие или отсутствие определенного ключа HASP HL.



## Методы защиты с помощью HASP HL

Система HASP HL предлагает защиту двумя методами:

- Осуществление защиты с помощью HASP HL Envelope.
- Осуществление защиты с помощью API системы HASP HL.

### HASP HL Envelope

Применение HASP HL Envelope является самым простым и быстрым способом защиты приложения. Envelope обеспечивает мощную и эффективную защиту исполняемых файлов или библиотек DLL. HASP HL Envelope осуществляет шифрование файла, добавляет проверки на наличие ключа HASP, а также встраивает мощные антиотладочные функции. После того, как вы защитили приложение с помощью HASP Envelope, его запуск становится невозможным без соответствующего ключа.

Поскольку HASP HL Envelope не требует доступа к исходному коду приложений, его основными преимуществами становятся быстрота и простота использования. В то же время Envelope обеспечивает высокий уровень защиты, делающий практически невозможным эффективное применение отладочных и реинжиниринговых средств.

Подробная информация о HASP HL Envelope приводится в разделе «Защита с помощью HASP HL Envelope».

## API системы HASP HL

Если вы располагаете исходным кодом приложения, которое вы намерены защитить, вы можете осуществить интеграцию системы HASP HL и этого приложения с помощью HASP HL API.

Используйте API для внедрения в приложение вызовов ключа HASP HL. API позволяет осуществлять проверку наличия ключа в любой выбранный момент и определять реакцию приложения в том случае, если требуемый ключ не подключен к компьютеру. Кроме того, можно осуществлять проверку памяти ключа HASP HL на предмет наличия записанных в нее данных.

## Какой метод выбрать?

Можно использовать либо один из методов (HASP HL Envelope или API), либо оба метода одновременно.

Применять HASP HL Envelope следует в том случае, если у вас нет доступа к исходному коду приложений, либо когда вы хотите защитить приложение максимально быстро и с минимальными сложностями.

Использовать API системы HASP HL следует в том случае, когда вы имеете доступ к исходному коду приложения, а также когда вы хотите самостоятельно настроить защиту программного обеспечения, определяя частоту и моменты проверки наличия соответствующего ключа HASP HL.

Как HASP HL Envelope, так и API являются мощными методами защиты. Защита приложения даже с помощью одного метода дает высокий уровень безопасности. Тем не менее, мы рекомендуем по возможности использовать оба метода. Каждый из них обладает собственными уникальными характеристиками и дополняет другой. Более подробная информация по данному вопросу содержится в разделе «Защита с помощью HASP HL — наилучшие подходы».

## Лицензирование с помощью HASP HL

Как уже было сказано ранее, лицензирование и защита приложения в системе HASP HL представляют собой разные процессы. Используя один ключ HASP HL можно лицензировать множество функций или приложений. Возможности лицензирования для различных моделей ключей HASP HL описаны в таблице 1.1.

---

## Возможности лицензирования

Система HASP HL позволяет контролировать использование защищенного приложения с помощью:

- a. установки ограничения на количество запусков определенных приложений или их отдельных функций. Эта возможность особенно важна при распространении демонстрационных версий программного обеспечения;
- b. установки даты окончания действия лицензии на определенную функцию или все приложение. Эта возможность особенно важна в случае аренды вашего программного обеспечения;
- c. установки максимального числа рабочих станций, на которых приложение может быть запущено одновременно. Эта возможность особенно важна в случае поставки версий программного обеспечения, предназначенных для всей компании клиента.

## Как работает лицензирование с помощью HASP HL?

Приложение, защищенное HASP HL, осуществляет несколько проверок.

- a. В первую очередь, приложение определяет наличие соответствующего ключа HASP HL.
- b. Если корректный ключ HASP HL подключен к компьютеру, осуществляется дальнейшая проверка с целью подтвердить авторизацию на запуск той или иной функции или всего приложения. Такая авторизация основывается на условиях лицензии.

Если приложение имеет авторизацию на запуск, в дальнейшем осуществляются дополнительные проверки, тип которых определяется моделью ключа HASP HL, используемого в защите приложения:

- в случае использования HASP HL Max или Pro, осуществляется проверка на превышение числа разрешенных запусков. С каждым запуском приложения количество разрешенных запусков уменьшается на единицу. В случае осуществления запуска приложения при числе разрешенных запусков, равном 0, защищенное приложение завершает работу. При этом появляется соответствующее сообщение об ошибке;
- в случае использования ключа HASP HL Time, проверяется дата истечения действия лицензии (хранится в ключе), которая затем сравнивается со значением внутренних часов ключа. Если дата истечения лицензии уже наступила, приложение или функция прекращает работу и появляется соответствующее сообщение об ошибке;

- в случае использования ключа HASP HL Net, защищенное приложение проверяет, не превышено ли разрешенное количество одновременных запусков приложения на сетевых рабочих станциях. Более подробная информация по этому вопросу содержится в разделе «HASP License Manager».

## **Система дистанционного перепрограммирования**

Лицензии заносятся на ключ HASP HL с помощью утилиты HASP HL Factory. Тем не менее, уникальная Система Дистанционного Перепрограммирования позволяет генерировать файлы, которые могут использоваться конечными пользователями для обновления лицензий непосредственно на месте работы защищенного приложения. Все ключи HASP HL (за исключением HASP HL Basic) могут перепрограммироваться дистанционно. У ключа HASP HL Basic отсутствует поддержка лицензий.

Система дистанционного перепрограммирования также позволяет получать информацию о текущем статусе лицензий, хранящихся на распространенных ключах HASP HL. Более подробная информация об этом содержится в разделе «Система дистанционного перепрограммирования».

## **Порядок работы с HASP HL**

В данном разделе представлен пример работы с системой HASP HL, иллюстрирующий концепцию Автоматической системы лицензирования. Разработчик программного обеспечения хочет защитить свой текстовый редактор. После обсуждения принимается решение лицензировать использование следующих функций:

- PRINT (Вывести на печать)
- SAVE (Сохранить)
- IMPORT (Импортировать)

Для выполнения описываемых ниже действий системой HASP HL предлагаются специальные инструменты.

1. Первоначально необходимо присвоить каждому лицензируемому компоненту или функции специальный идентификатор (Feature ID). Создается перечень функций (как показано в таблице ниже), который в дальнейшем передается разработчикам.

**Таблица 2.1 Пример перечня функций**

Функция	Идентификатор
PRINT	34
SAVE	4
IMPORT	48

- На основании перечня разработчик реализует защиту. Разработчик знает только идентификаторы. Нет необходимости информировать его о коммерческих вопросах или вопросах лицензирования, и даже о типе ключа HASP HL, который планируется использовать. Например, разработчик использует API в целях реализации лицензирования для функции с идентификатором 34. Разработчику нет нужды знать, за что будет в дальнейшем платить пользователь — за использование определенное количество раз функции печати, за возможность осуществлять печать до определенной даты, или за определенное количество работающих одновременно пользователей. Эти вопросы решаются бизнес-подразделением и на данной стадии реализации защиты не имеют значения. Разработчик просто реализует защиту с использованием идентификатора в качестве одного из параметров защиты.
- Бизнес-подразделение определяет дистрибуционные пакеты и условия их лицензирования. Пакет содержит одну или более функций и условия их лицензирования. В таблице 2.2 приводится пример пакетов, их функции и особые условия лицензирования.

**Таблица 2.2 Пример перечня пакетов**

Название пакета	Функции, включенные в пакет	Условия лицензии	Используемая модель ключа HASP HL
Демонстрационный	PRINT SAVE IMPORT	30 дней для каждой функции	HASP HL Time
Стандартный	PRINT SAVE	Без ограничения	HASP HL Pro
Корпоративный	PRINT SAVE IMPORT	Без ограничения	HASP HL Pro
Профессиональный	PRINT SAVE IMPORT	До 25 работающих одновременно пользователей	HASP HL Net

4. Теперь бизнес-подразделение может отдать указание менеджерам по продажам начинать продажи продукта и оформлять заказы. Департамент заказов/производства получает заказы на определенный пакет. Одним щелчком мыши работник департамента выбирает соответствующий пакет и желаемый ключ HASP HL, который соответствующим образом программируется.
5. На более поздней стадии клиент может запросить обновления. Обновление может представлять собой продление существующей лицензии или лицензию на дополнительные модули. Сотрудник департамента заказов/производства получает от клиента информацию о требуемом обновлении и выбирает соответствующий пакет, который в дальнейшем трансформируется в данные для перепрограммирования и посылается клиенту.

 HASP HL подходит и клиентам, которые не нуждаются в лицензировании. В этом случае разработчик может просто реализовать защиту, в основе которой будут лежать проверки наличия соответствующего ключа HASP HL.

Рисунок 2.3 Пример работы Автоматического лицензирования



# Установка HASP HL

Эта глава предназначена для того, чтобы помочь вам начать работать с системой HASP HL. В главе рассматриваются следующие темы:

- краткий обзор существующего программного обеспечения HASP HL;
- краткое описание реализации защиты с помощью HASP HL;
- процесс установки HASP HL на ваш компьютер;
- процесс установки драйверов устройств HASP HL;
- работа с мастер-ключами HASP HL на вашей системе.

## Существующее программное обеспечение HASP HL

Для установки приведенных ниже компонентов следует использовать установочный компакт-диск HASP HL:

- Драйверы и демоны, используемые для обмена данными с ключами HASP HL.
- Набор утилит **Vendor Center**, который применяется для защиты и лицензирования вашего программного обеспечения.
- API системы HASP HL и его библиотеки, которые используются для реализации защиты вашего программного обеспечения. На компакт-диске также содержатся примеры на различных языках программирования.
- HASP License Manager. Используется для обмена данными между защищенным приложением и сетевыми ключами HASP HL Net.
- Aladdin Monitor. Используется для отслеживания использования защищенного приложения в сетевой среде.
- Aladdin DiagnostiX и Aladdin DiagnostiX Memory Beamer. Используются для устранения ошибок в работе ключей.

## Драйверы HASP HL

Драйверы устройств HASP HL служат для обеспечения связи между HASP HL и защищенным приложением после установки последнего на компьютере клиента. Для работы с защищенным приложением конечному пользователю потребуется установить соответствующие драйверы устройств.

Драйверы устройств HASP HL предназначены для работы со следующими операционными системами: Windows 98 SE/ME/2000/XP/Server 2003, Mac и Linux.

Установку драйверов устройств HASP HL можно интегрировать в установку защищаемого приложения. Для Windows XP и Windows Server 2003 драйверы устройств HASP HL можно получить с помощью Windows Update.

## Vendor Center

Пакет утилит Vendor Center содержит следующие программы:

- **HASP HL Envelope.** Создает защитную оболочку вокруг исполняемых файлов защищаемого приложения.
- **HASP HL Factory.** Используется для создания лицензий для защищаемого приложения, а также для инициализации памяти ключей HASP HL.
- **HASP HL ToolBox.** Используется для изучения API системы HASP HL, а также для генерирования программного кода, включаемого для защиты в исходный код приложения.

## API системы HASP HL

API системы HASP HL позволяет защитить ваше приложение с помощью размещения в разных частях исходного кода вызовов ключа HASP HL.

Корневой каталог установочного компакт-диска HASP HL имеет поддиректории, соответствующие разным операционным системам. В каждой поддиректории есть папка Sample. Папка **Sample** содержит примеры для различных компиляторов и языков программирования. Каждый пример имеет:

- Библиотеки, которые должны быть включены в приложение
- Пример приложения, демонстрирующий использование API.

## HASP License Manager/Aladdin Monitor

В случае если приложение связано с ключом HASP HL Net, работающим в сетевой среде, следует использовать обе утилиты.

## Утилиты, предназначенные для конечных пользователей

К таким утилитам относятся Aladdin DiagnostiX и Aladdin DiagnostiX Memory Weaver. Более подробно применение данных утилит описывается в разделе «Диагностика ключей HASP HL»

---

## Краткое описание защиты программного обеспечения с помощью HASP HL

Раздел представляет собой краткое описание шагов, необходимых для защиты приложения с помощью HASP HL. Как вы увидите, ваше программное обеспечение может быть защищено в течение нескольких минут. Для демонстрации возможностей HASP HL вам потребуется:

- Ваш установочный компакт-диск HASP HL
- Демонстрационный ключ HASP HL, входящий в состав Демонстрационного Комплекта.

### 1. Установите программное обеспечение HASP HL

Вставьте установочный компакт-диск и следуйте появляющимся на экране инструкциям, пока не появится подсказка, информирующая вас о том, что установка была успешной.

### 2. Подключите аппаратные средства

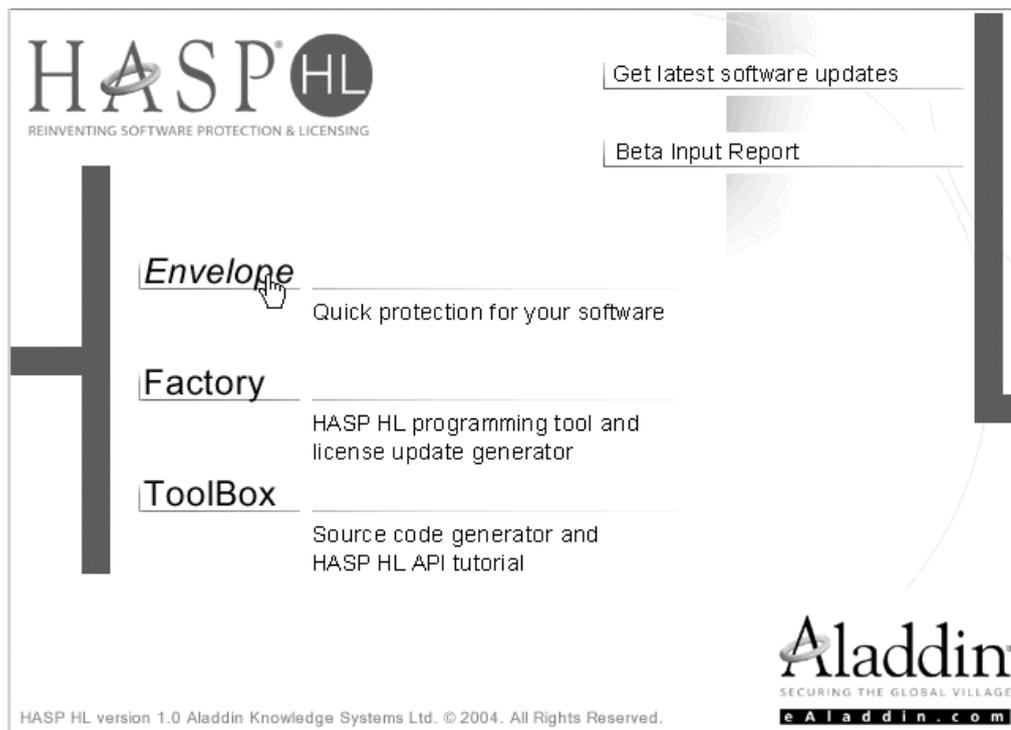
Подключите демонстрационный ключ HASP HL к порту USB вашего компьютера. Данный ключ следует использовать только в целях оценки возможностей системы.

### 3. Запустите Vendor Center

В меню **Start** операционной системы Windows выберите **Programs/HASP HL/Vendor Center**

На экране появится окно утилиты Vendor Center (см. рисунок 3.1).

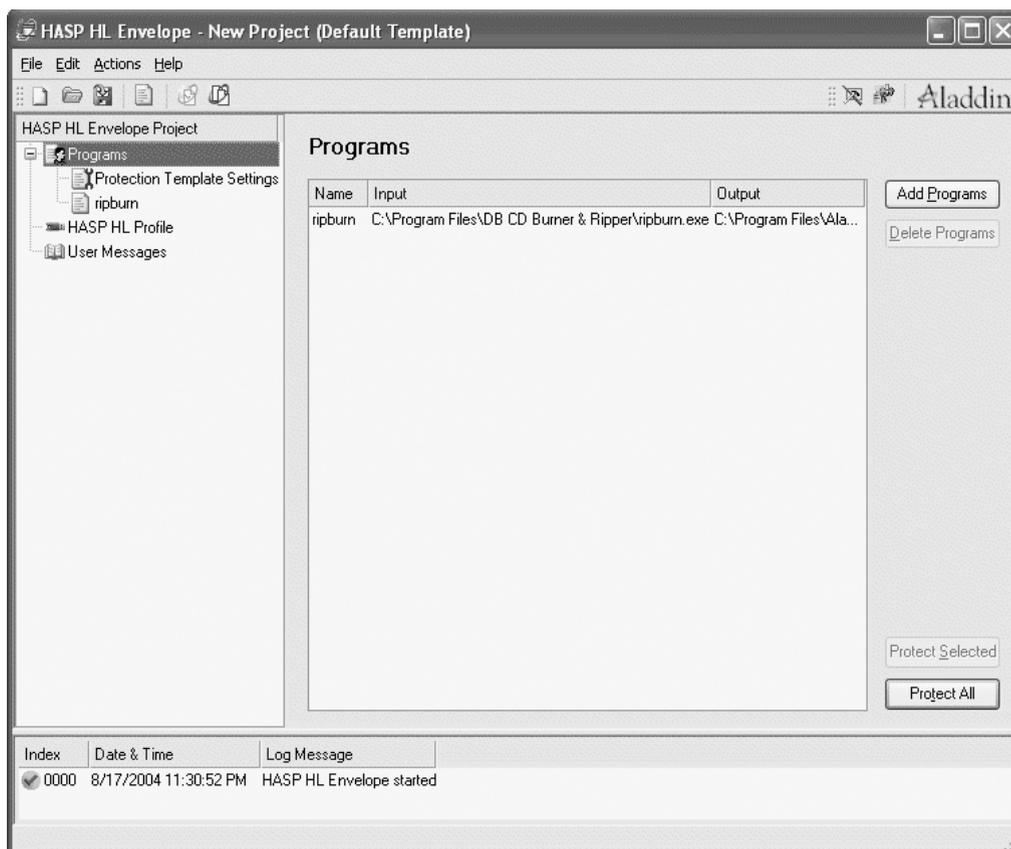
Рисунок 3.1 Утилита HASP HL Vendor Center



#### 4. Выберите HASP HL Envelope

Появится окно, соответствующее рисунку 3.2. HASP HL Envelope является инструментом, который применяется для создания быстрой автоматической защиты.

Рисунок 3.2 Окно HASP HL Envelope

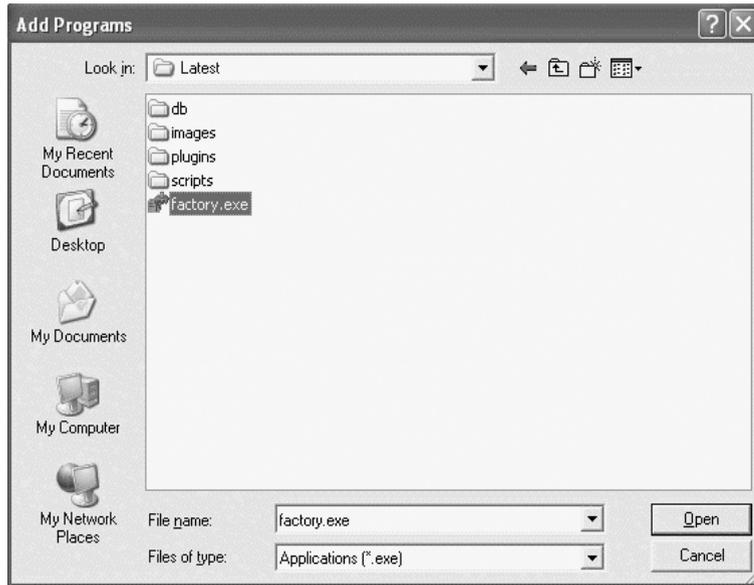


## 5. Выберите объект защиты

- a. Щелкните на символе **Programs** в области **HASP HL Envelope Project**.
- b. Выберите исполняемый файл, который вы хотите защитить с помощью HASP HL. Перетащите файл в область **Programs**, либо вы можете щелкнуть по кнопке **Add Programs** в области **Programs**.

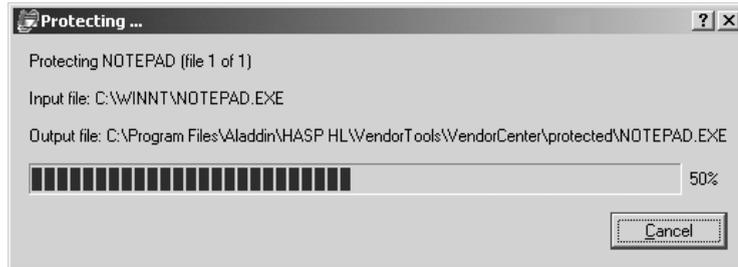
Появится диалоговое окно **Add Programs**, показанное на рисунке 3.3. В нем следует выбрать файл, который вы хотите защитить.

Рисунок 3.3 Диалоговое окно Add Programs

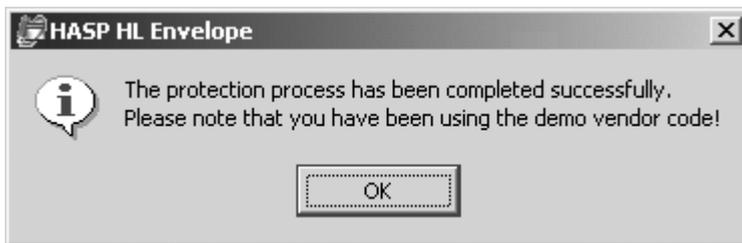


Выберите файл, который вы хотите защитить и щелкните по кнопке **Open**. Диалоговое окно закроется. Выбранная программа появится в области **Programs**.

- c. Для защиты файла щелкните по кнопке **Protect All** в области **Programs**. Появится окно с индикатором выполнения задачи:



После того, как процесс создания защиты завершится, появится следующее сообщение:



## 7. Протестируйте защищенный файл

- a. Запустите защищенное приложение. Убедитесь в том, что ключ HASP HL Demo подключен к компьютеру.
- b. Закройте защищенное приложение.
- c. Отключите ключ HASP HL Demo.

Повторно запустите защищенное приложение. На экране появится приведенное ниже сообщение об ошибке.



Защищенное приложение не будет работать без подключенного ключа HASP HL!

## Установка HASP HL в среде Windows

Первым этапом в установке системы HASP HL является установка всего необходимого программного обеспечения. В этом разделе разъясняется процесс установки программного обеспечения в операционной системе Windows. Среда разработчика HASP HL и утилиты Vendor Center могут быть установлены на системы с Windows 2000 и более поздними версиями Windows.

### Установка программного обеспечения HASP HL

Вставьте установочный компакт-диск HASP HL. Автоматически запустится мастер установки HASP HL, а на экране появится приветственное окно (рис. 3.4).

Рисунок 3.4 Мастер установки HASP HL



Для установки содержимого компакт-диска следуйте появляющимся на экране инструкциям.



Если по какой-либо причине установка не происходит в автоматическом режиме, следует запустить файл setup.exe из папки Windows установочного компакт-диска.

---

## Структура устанавливаемых компонентов

При обычной установке под операционную систему Windows в папке HASP HL будут созданы следующие папки:

- **API** — содержит библиотеки API.
- **Drivers** — содержит драйверы устройств HASP HL (более подробно описываются в следующем разделе).
- **Samples** — содержит примеры приложений для различных языков программирования и интерфейсов.
- **Vendor Tools** — содержит набор инструментов Vendor Center и Aladdin Monitor.
- **Docs** — содержит версию данного руководства, а также дополнительную документацию по применению системы HASP HL.
- **Redistribute** — содержит программное обеспечение, которое следует устанавливать на компьютерах конечных пользователей. Более подробная информация по этому программному обеспечению содержится в разделе «Программное обеспечение для конечных пользователей HASP HL».



Описанная выше структура устанавливаемых компонентов повторяется установочным компакт-дискom в папке **Windows/Installed**.

## Установка драйверов устройств HASP HL

Драйверы устройств HASP HL устанавливаются автоматически при запуске программы установки HASP HL.

Возможна также отдельная установка драйверов, с помощью запуска *haspdinst.exe* или *HASPUserSetup.exe*. Эти приложения находятся в папке **Drivers** вашего установочного диска HASP HL.

Приложения *haspdinst.exe* и *HASPUserSetup.exe* распознают используемую операционную систему и устанавливают соответствующие ей файлы драйверов.



Для установки драйверов устройств в операционных системах Windows 2000/XP/ Server 2003 необходимы права администратора.

Драйверы HASP HL загружаются динамически при подключении ключа HASP HL.

### Приложение HASPUserSetup.exe

Эта программа с графическим интерфейсом предназначена для отдельной установки драйверов HASP HL на компьютеры с операционными системами Windows 98/2000/XP/Server 2003. Исполняемый файл находится в папке установочного компакт-диска HASP HL по следующему пути: `Windows/Installed/Drivers`.

Для запуска приложения следует выбрать файл *HASPUserSetup.exe* и следовать появляющимся на экране инструкциям.

### Утилита `haspdinst.exe`

Утилита *haspdinst.exe* не имеет графического интерфейса (только командная строка) и также предназначена для установки драйверов устройств в системах Windows 98/2000/XP/Server 2003.

#### **Для установки драйверов устройств HASP HL**

Напечатайте в командной строке `haspdinst -i`

Появится сообщение, информирующее вас о том, что драйверы устройств HASP HL были успешно установлены.

#### **Для удаления драйверов устройств HASP HL**

Напечатайте в командной строке `haspdinst -r`

Появится сообщение, информирующее вас о том, что драйверы устройств HASP HL были успешно удалены.

#### **Для установки новой версии драйверов устройств HASP HL**

Установите более новые драйверы устройств HASP HL в соответствии с процедурой, описанной выше. Утилита *haspdinst.exe* автоматически осуществит процесс обновления.

### Прочие команды утилиты *Haspdinst*

Прочие команды, которые могут быть выполнены утилитой *haspdinst.exe*, указаны в таблице 3.1.

**Таблица 3.1** Команды утилиты *Haspdinst*

Команда	Описание
-info	Показывает статус установки.
-h и -?	Показывает список доступных команд.
-kr	Позволяет программе установки прекратить все процессы, обращающиеся к драйверам.
-cm	Указывает программе установки показывать только критические сообщения (например, просьбу осуществить перезагрузку).
-fr	Дает указание программе установки принудительно удалить драйвер.
-nomsg	Дает указание программе установки не выводить никаких сообщений.

## Установка HASP HL в операционной системе Mac

Программное обеспечение для Mac-систем находится в папке *MacOS* установочного компакт-диска HASP HL. Папка включает следующие подкаталоги:

- **API** — содержит файлы заголовков и библиотеки для различных языков программирования.
- **Docs** — содержит электронную версию документации по применению системы HASP HL.
- **Redistribute** — содержит программное обеспечение, которое следует устанавливать на компьютерах конечных пользователей.



Данная папка также содержит файлы для установки демона HASP HL.

- **Samples** — содержит примеры программ для различных языков программирования с реализацией API системы HASP HL.
- **VendorTools** — содержит версию HASP HL Envelope с интерфейсом командной строки. Более подробная информация по использованию данного приложения находится в разделе «HASP HL Envelope для приложений Mac».

Содержимое данных папок не устанавливается на ваш компьютер автоматически. Его необходимо скопировать с установочного компакт-диска самостоятельно.

## Установка демона HASP HL

Для того чтобы ваша система распознала и могла нормально обмениваться данными с аппаратными средствами HASP HL, необходимо установить *aksusbd* — демон HASP HL.

### Установка демона

1. Подключите установочный образ HASP *Installation.dmg*, который находится по следующему пути: *MacOS/Redistribute/Runtime/dmg*. Будет подключен том, содержащий 4 файла.
2. Откройте файл *AKSUSB Install.pkg*. Будет запущена встроенная программа установки (см. рис. 3.5), которая осуществит процесс инсталляции.

Рисунок 3.5 Экран HASP HL Mac Installer



Описанный выше процесс установки требует, чтобы на вашей системе была установлена подсистема BDS. Если она не установлена, следует использовать файл *AKSUSB Install\_forced.pkg*. Такой вариант требует перезагрузки после завершения установки.

### Что необходимо для корректной работы

Перед запуском демона на вашем компьютере необходимые компоненты должны быть установлены следующим образом:

- a. Демон должен находиться на следующем томе: `/usr/libexec/`
- b. Два загрузочных файла, обеспечивающих активацию демона после перезагрузки, должны находиться на следующем томе: `System/Libraries/StartupItems/Aladdin`



Когда все компоненты будут корректно установлены, индикатор на ключе HASP HL будет гореть.

## Опции *aksusbd*

### Общие изменения

В работу демона можно внести изменения с помощью ключей командной строки, описанных в таблице 3.2.

### Процесс изменения

Для внесения временных изменений в работу демона:

1. Запустите программу **Activity Monitor**, которая находится по следующему пути: `/Applications/Utilities/`
2. Выберите из списка, показываемого при просмотре **All Processes**, процесс **aksusbd**.
3. Завершите процесс – для этого необходимы права администратора.
4. Откройте окно оболочки и запустите демон с измененными опциями.

Чтобы убедиться в том, что измененные настройки после перезагрузки вступили в силу, следует добавить сделанные изменения в скрипт оболочки Aladdin, который находится по следующему пути:  
`/System/Libraries/StartupItems/Aladdin/`

Таблица 3.2 Параметры командной строки для *aksusbd* (Mac)

Ключ	Значение
-v	Напечатать номер версии в десятичном формате, формат xx.xx
-l <значение>	Выбрать тип диагностических сообщений. Возможные значения: 0 — только ошибки 1 — обычные (по умолчанию) 2 — с пояснениями 3 — с полными пояснениями Сообщения заносятся в системный журнал с приоритетом <i>kern.info</i> (и <i>kern.debug</i> ). Для того чтобы выяснить, где хранятся сообщения, следует обратиться к <i>/etc/syslog.conf</i> . Обычно они хранятся в <i>/var/log/messages</i> .
-u <маска>	Определяет биты разрешения к специальному файлу сокетов. Значение по умолчанию — 666 (доступ для всех пользователей).
-d	Устанавливает задержку времени в миллисекундах для доступа к ключу HASP HL. Значение по умолчанию — 250.
-h	Показывает сведения о приложении, запускаемом из командной строки.

## Установка HASP HL в операционной системе Linux

Программное обеспечение HASP HL для операционной системы Linux включает следующие компоненты:

- демон HASP HL;
- исходный код приложения HASP HL Demo;
- библиотеки HASP HL.

Программное обеспечение для Linux находится в папке Linux установочного компакт-диска HASP HL.

Самым простым способом установки HASP HL на компьютере с операционной системой Linux является использование RPM-пакетов, находящихся на установочном компакт-диске. Программное обеспечение HASP HL для различных версий Linux постоянно обновляется на сайте <http://www.hasp.com/downloads>.

Подробная информация об отдельных компонентах находится в файле *readme.txt* в папке *Linux*.

### Установка демона HASP HL

Для получения доступа к ключу HASP HL необходимо загрузить демон *aksusbd*.



Все действия, описанные в данном разделе, должны выполняться как root.

### Установка *aksusbd* с использованием RPM-пакетов

На установочном компакт-диске содержатся RPM-пакеты для различных версий SuSE и Red Hat. Эти пакеты позволяют ускорить процесс установки.

Установка демона HASP HL с использованием RPM-пакетов:

1. В папке *Linux* выберите необходимый пакет.
2. Введите команду установки, используя следующий формат:  
`rpm -i [имя пакета]`



Для получения полного списка доступных для RPM-пакетов команд следует использовать следующую команду: `man rpm`

---

## Установка демона Aladdin (*aksusbd*)

### Разрешение доступа к USB-ключам

Для того чтобы разрешить доступ к USB-ключам, *subdues* должен быть подключен к */proc/bus/usb*. В более новых версиях (например, SuSE 8.0) он подключается автоматически.

Для ручного подключения *usbdevfs* следует использовать следующую команду:

```
Mount -t usbdevvfs none /proc/bus/usb
```

### Загрузка демона

Загрузите демон, запустив следующую команду:

```
<путь>/aksusbd
```

Демон запустится и перейдет в фоновый режим.

В системном журнале генерируется сообщение о статусе, информирующее об успешности или ошибках при установке. В сообщении содержится версия демона, а также версия API системы HASP HL, используемая для USB.

Если */proc/bus/usb* не подключен при запуске *aksusbd*, к USB-ключам доступ будет невозможен.

Рекомендуется, чтобы демон запускался при загрузке системы и чтобы скрипты находились по следующему пути: */etc/rc.d/init.d* или */etc/init.d*

**Опции aksusbd**Таблица 3.3 Ключи командной строки для *aksusbd* (Linux)

Ключ	Значение
-v	Напечатать номер версии в десятичном формате, формат xx.xx
-l <значение>	Выбрать тип диагностических сообщений. Возможные значения: 0 – только ошибки 1 – обычные (по умолчанию) 2 – с пояснениями 3 – с полными пояснениями Сообщения заносятся в системный журнал с приоритетом <i>kern.info</i> (и <i>kern.debug</i> ). Для того чтобы выяснить, где хранятся сообщения, следует обратиться к <i>/etc/syslog.conf</i> . Обычно они хранятся в <i>/var/log/messages</i> .
-u <маска>	Определяет биты разрешения к специальному файлу сокетов. Значение по умолчанию – 666 (доступ для всех пользователей).
-d	Устанавливает задержку времени в миллисекундах для доступа к ключу HASP HL. Значение по умолчанию – 250.
-h	Показывает сведения о приложении, запускаемом из командной строки.

---

## Получение Кодов Разработчика

При заказе ключей HASP HL вам присваивается уникальный Код Разработчика. Ключи HASP HL будут доставлены вам вместе с мастер-ключом HASP HL. Обзор взаимосвязи между серий ключа и вашими ключами приведен в разделе «Ваши уникальные коды и ключи HASP HL».

Доставленный вам мастер-ключ HASP HL содержит ваш уникальный Код Разработчика, а также иную информацию, используемую системой HASP HL для защиты и лицензирования программного обеспечения.

Для того чтобы получить необходимую вам информацию с мастер-ключа HASP HL:

1. подключите ключ Master HASP HL к вашему компьютеру;
2. запустите любой инструмент из состава **Vendor Center (HASP HL Envelope, HASP HL Factory** или **HASP HL ToolBox**);
3. утилита определит наличие нового мастер-ключа HASP HL;



Существует возможность указать Access Code (код доступа) к мастер-ключу HASP HL. Чтобы задать код, щелкните по кнопке **Change**. В появившемся окне введите код в соответствующем поле. Этот код понадобится вам при использовании HASP HL Factory для лицензирования защищаемого программного обеспечения. Щелкните по кнопке **OK**, чтобы закрыть окно.

4. выберите имя файла, под которым вы хотите сохранить информацию о коде разработчика. Настоятельно рекомендуется хранить все Коды Разработчика в папке со следующим путем:  
*Program Files/Aladdin/HASP HL/Vendorcodes/*

Инструменты из набора утилит Vendor Center по умолчанию обращаются именно к этой папке.



# Защита программного обеспечения

В данной главе рассматриваются следующие темы, описывающие защиту программного обеспечения с помощью HASP HL:

- резюме по принципам работы защиты с помощью HASP HL;
- основные аспекты защиты с помощью HASP HL;
- Введение в методы защиты с помощью HASP HL.

В последующих главах различные методы защиты с помощью HASP HL и подходы к повышению защищенности программного обеспечения с помощью данной системы будут описываться более подробно.

Раздел «Защита с помощью API системы HASP HL» подробно описывает метод защиты с использованием API.

Раздел «Защита с помощью HASP HL Envelope» подробно описывает метод защиты с использованием этого инструмента.

Стратегии и рекомендации по повышению защищенности программного обеспечения с помощью HASP HL описываются в разделе «Стратегии защиты».

## Защита с помощью HASP HL

Система HASP HL представляет собой современное программно-аппаратное средство защиты приложений от незаконного или неавторизованного использования. Система предотвращает незаконный доступ к защищаемому приложению или его выполнение.

Установленная программа, защищаемая HASP HL, требует наличия определенного ключа HASP HL. Защищаемая программа запрашивает с ключа специфическую информацию. Если ключ HASP HL отсутствует или получаемая информация с ключа вызывает подозрения, программа либо не запускается, либо прекращает работу.

После выбора метода защиты с помощью HASP HL следует стадия его реализации. Вне зависимости от выбранной стратегии защиты, защищаемые программы работают корректно только при получении особой информации с определенного ключа HASP HL.

### Элементы защиты HASP HL

В основе защиты с помощью HASP HL лежат следующие компоненты:

- идентификация соответствующего ключа HASP HL;
- защита программ и файлов данных;
- использование крипто-процессора и алгоритма AES;
- секретные параметры защиты;
- антиотладочные и антиреинжиниринговые средства защиты;
- использование памяти ключей HASP HL .

### Идентификация ключа HASP HL

Ключ HASP HL является аппаратным компонентом системы защиты HASP HL.

Вне зависимости от используемого метода защиты, защищенные программы работают только в том случае, если они могут получить доступ к необходимой информации, содержащейся на определенном ключе. Ключ HASP HL и его содержимое не может быть воспроизведен хакерами в целях раскрытия связи между программным обеспечением и аппаратными средствами.

Основным фактором защиты HASP HL является возможность защищаемой программы в процессе собственной работы идентифицировать и получить затем доступ к данным, хранящимся на определенном ключе. Этот фактор четок и ясен: ключ либо есть, либо его нет!

## Методы защиты с помощью HASP HL

HASP HL предусматривает использование двух основных методов:

- HASP HL API;
- HASP HL Envelope.

При защите программ с помощью обоих методов между приложениями и определенным ключом HASP HL формируется неразрывная связь.

### Что может быть защищено?

Система HASP HL предлагает в данном вопросе широкий выбор. Вы можете использовать систему для защиты:

- Скомпилированных исполняемых файлов и библиотек DLL.
- Определенных функций или всей программы целиком — HASP HL защищает все уровни программного обеспечения. Защита с помощью HASP HL может реализовываться как на функциональном уровне, так и на уровне всей программы.
- Конфиденциальных данных или интеллектуальной собственности.

Все вышеперечисленные компоненты будут защищены от любой попытки реинжиниринга.

Возможные параметры защиты, создаваемой с помощью HASP HL, описываются в следующих двух главах.

## Шифрование с помощью алгоритма AES

Защищаемая программа в своем функционировании зависит от определенного ключа HASP HL. В дополнение к проверкам наличия ключа предусмотрена возможность шифрования и дешифрования данных с использованием HASP HL.

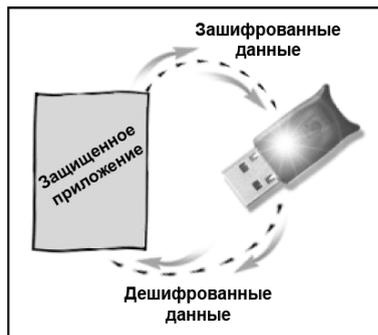
### Шифрование и дешифрование с использованием алгоритма AES

Встроенный крипто-процессор ключей HASP HL осуществляет криптографические операции на основе алгоритма AES. При осуществлении шифрования HASP HL использует набор секретных 128-битных ключей, которые не покидают памяти аппаратного ключа HASP HL.

Реализуемые вами схемы защиты должны быть более сложными, чем просто запрос на наличие определенного ключа HASP HL. Однако проверка наличия конкретного ключа HASP HL с помощью операций шифрования/дешифрования требует планирования. Изначально необходимы зашифрованные данные, которые в дальнейшем посылаются на ключ, где и происходит их дешифрование.

Если данные верны, система рассматривает это как наличие соответствующего ключа HASP HL (см. рисунок 4.1). Более подробная информация содержится в разделе «Функции шифрования».

**Рисунок 4.1** Шифрование/дешифрование данных



## Параметры защиты конфиденциальной информации

Суть защиты программного обеспечения заключается в сохранении конфиденциальности. Без конфиденциальных элементов любая система защиты ПО подвергается риску.

### Уникальный Код разработчика

Каждому покупателю системы HASP HL присваивается уникальный Код Разработчика. Этот код должен оставаться конфиденциальным.

Код Разработчика является неотъемлемой частью защитных параметров, которые образуют связь между защищенными программами и HASP HL. Тем не менее, Код Разработчика представляет собой только часть такой связи, а не всю связь целиком. Сам по себе этот код недостаточен для получения незаконного доступа к программному обеспечению. Он лишь позволяет защищаемому приложению получить доступ к ключу и его ресурсам.

Все инструменты HASP HL, предназначенные для создания защиты, требуют Код Разработчика. Информация по получению этого кода содержится в разделе «Получение доступа к Кодам Разработчика».

## Использование памяти ключа HASP HL

Большинство моделей аппаратных ключей HASP HL оснащено защищенной памятью. Память ключа (доступна на чтение и запись) может использоваться в качестве части схемы защиты программного обеспечения. В памяти ключа могут храниться конфиденциальные дан-

ные, включая части программного кода, имя клиента или любые иные данные.

Для записи данных в память ключа HASP HL следует использовать редакторы памяти, входящие в состав HASP HL ToolBox. Более подробная информация по этому вопросу содержится в разделе «Функции памяти».

## **Антиотладочные и антиреинжиниринговые средства**

Система HASP HL защищает интеллектуальную собственность и укрепляет защиту с помощью антиотладочных и антиреинжиниринговых средств. С помощью средств взлома хакеры обычно пытаются раскрыть схему защиты приложения, осуществляя трассировку скомпилированного приложения к его исходному коду. Средства HASP HL Envelope позволяют создать защиту от подобных атак, а также предотвратить возможность раскрытия алгоритмов, используемых внутри защищенного программного обеспечения.

## **Выбор метода защиты**

Как было указано ранее в разделе «Методы защиты с помощью HASP HL», данная система позволяет использовать два варианта защиты: HASP HL API и HASP HL Envelope. Оба метода позволяют создать связь между защищенным программным обеспечением и содержимым определенного ключа HASP HL.

При выборе метода защиты необходимо принять во внимание следующие вопросы:

- Что будет защищать HASP HL?
- Как будут применяться параметры защиты HASP HL?
- Временной фактор — требуется ли автоматическое и быстрое решение?
- Важна ли гибкость в реализации схемы защиты?

### **Что защищать?**

При создании защиты программного обеспечения с помощью HASP HL существует несколько вариантов ее реализации. HASP HL API используется для защиты программного обеспечения до его компиляции. Защита может быть реализована и с помощью HASP HL Envelope — в этом случае уже после того, как программное обеспечение было скомпилировано. Вы можете выбрать конкретный объект защиты — всю программу, подпрограмму, или просто функцию.

## Как применить защиту

При применении HASP HL API защита интегрируется на уровне исходного кода. Решение о том, в каком месте исходного кода следует разместить вызовы HASP HL API, принимается вами. HASP HL Envelope представляет собой более автоматизированный способ защиты программного обеспечения. Применяемые в защищенной программе параметры защиты определяются вами.

## Важен ли контроль над схемой защиты?

При применении HASP HL API вами контролируется весь процесс создания защиты. Вами задается время вызова приложением ключа HASP HL, а также реакция приложения в различных ситуациях. В случае применения HASP HL Envelope параметры защиты при создании оболочки вокруг скомпилированного файла носят случайный характер. Если дважды использовать HASP HL Envelope для защиты одной и той же программы, в результате получается два файла с разными защитными модулями и оболочками.

## Применение HASP HL API в сравнении с применением HASP HL Envelope

В зависимости от ваших потребностей, вы можете использовать любой из методов защиты либо скомбинировать оба метода.

**Таблица 4.1 HASP HL API в сравнении с HASP HL Envelope**

HASP HL Envelope	HASP HL API
<ul style="list-style-type: none"> <li>Автоматический процесс защиты и определение специфических параметров защиты программ.</li> </ul>	<ul style="list-style-type: none"> <li>Ручная реализация вызовов HASP HL API.</li> </ul>
<ul style="list-style-type: none"> <li>Простая и быстрая защита программного обеспечения.</li> </ul>	<ul style="list-style-type: none"> <li>Контролируемый и кропотливый процесс, обеспечивающий максимальную защиту. Степень защищенности зависит от объема функций API, реализованных при создании защиты.</li> </ul>
<ul style="list-style-type: none"> <li>Исходный код не требуется</li> </ul>	<ul style="list-style-type: none"> <li>Исходный код необходим.</li> </ul>
<ul style="list-style-type: none"> <li>Антиотладочные и антиреинжиниринговые средства.</li> </ul>	<ul style="list-style-type: none"> <li>Максимальная гибкость.</li> </ul>

# Защита с помощью системы HASP HL API

В главе рассматриваются следующие основные темы, связанные с применением для защиты приложений HASP HL API:

- обзор возможностей HASP HL API;
- условия, необходимые для применения API;
- обучение применению API;
- реализация защиты с помощью API;
- доступные функции API.

## Обзор

API системы HASP HL представляет собой мощное средство защиты программного обеспечения, эффективность которого полностью зависит от того, как вы его используете. Чтобы достичь наилучшего результата от применения API, реализация этого средства должна быть максимально сложной.

В справочнике по HASP HL API (Приложение С данного документа) подробно описываются составляющие его функции. Для получения дополнительной информации по конкретным интересующим вас функциям следует обращаться именно к этому приложению.

Для защиты вашего приложения с помощью HASP HL API в исходный код защищаемого приложения следует вставить вызовы ключа HASP HL. Вы можете вставить проверки наличия ключа в любое место приложения, а также определить его реакцию в зависимости от полученного ответа. Если требуемый ключ не обнаружен, приложение может прекратить работу или перейти в демонстрационный режим.

Использование HASP HL API позволяет проверить память аппаратного ключа на наличие определенных данных, а также осуществлять шифрование и дешифрование данных.

Степень использования функций API определяет общий уровень защищенности приложения. Таким образом, необходимо, чтобы до реализации защиты вы ознакомились с возможностями HASP HL API.

Для ускорения процесса обучения мы рекомендуем использовать утилиту HASP HL ToolBox. Также рекомендуется ознакомиться с содержанием папки примеров, соответствующей вашему компилятору, по реализации HASP HL API. Каждый интерфейс HASP HL включает файл заголовка и пример приложения, демонстрирующий применение API. Примеры приложений находятся в папке Sample внутри каталогов Windows, Mac OSX и Linux установочного компакт-диска HASP HL.

## Универсальный API

Следует подчеркнуть, что API системы HASP HL является универсальным для всех моделей ключей HASP HL и для всех основных платформ.

С точки зрения API, модель аппаратного ключа не имеет значения. В отдельных моделях ключей HASP HL существует ряд особых функций, однако реализация защиты не зависит от типа ключей, связанных с защищаемыми приложениями.

Использование API также не зависит от режима получения доступа к ключу HASP HL. Те же функции API используются при программировании приложений на получение доступа как к ключу, находящемуся на какой-либо сетевой станции, так и к ключу, подключенному к данному компьютеру.



Реализация API системы HASP HL не зависит от модели ключа HASP HL или платформы, на которую устанавливается защищаемое приложение.

API системы HASP HL также универсален с точки зрения рабочей среды, в которой устанавливается защищаемое приложение. HASP HL является кросс-платформенной системой, поддерживающей операционные системы Windows, Mac и Linux.

## Условия, необходимые для применения API

Драйверы устройств или демон HASP HL должны быть установлены до того, как вы начнете использовать HASP HL API. Подробные сведения

---

об этом содержится в разделе «Установка программного обеспечения HASP HL».

## Требования к платформе

API системы HASP HL предназначен для работы на:

- платформах Windows 98SE и более новых версиях;
- Mac OSX;
- Linux.

## Код Разработчика

Для получения доступа к ресурсам и памяти ключа HASP HL необходим Код разработчика. Коды разработчика обычно хранятся в каталоге *Vendorcodes*. При установке под Windows данный каталог находится по следующему пути:

`Program Files/Aladdin/HASP HL/Vendorcodes.`

Покупателям HASP HL обычно предоставляются ключи HASP HL Demo, работающие с Кодом разработчика DEMOMA. Этот Код разработчика может использоваться при создании защиты с помощью HASP HL API.



Не распространяйте программное обеспечение, защищенное демонстрационным ключом HASP HL Demo. Этот ключ предназначен исключительно для оценки возможностей аппаратных средств.

После получения мастер-ключа HASP HL вам следует получить с него присвоенный вам Код разработчика.

Если вы уже использовали свой мастер-ключ HASP HL при работе с утилитами **Vendor Center**, следующий ниже раздел можно пропустить.

## Получение Ключа разработчика с мастер-ключа HASP HL

Для извлечения кода разработчика с мастер-ключа HASP HL необходимо осуществить следующие действия:

1. Подключите мастер-ключ к вашему компьютеру.
2. Запустите любой инструмент из состава **Vendor Center (HASP HL Envelope, HASP HL Factory** или **HASP HL ToolBox**).
3. Мастер определяет и выводит список всех мастер-ключей HASP HL.



Вы можете задать код доступа (Access Code) к мастер-ключу Master HASP HL. Чтобы сделать это:

- a. Щелкните по кнопке **Change**.
- b. В открывшемся окне введите в соответствующей области желаемый код. Этот код понадобится вам при использовании HASP HL Factory для лицензирования защищаемого программного обеспечения.
- c. Щелкните **OK**, чтобы закрыть окно.

4. Введите имя файла, под которым вы хотите сохранить код разработчика. Хранить все коды разработчика рекомендуется в директории *Vendorcodes*. На платформах с любой версией Windows данный каталог находится по следующему пути:  
`Program Files/Aladdin/HASP HL/Vendorcodes`.

## Обучение применению HASP HL API

Существует два способа изучения принципов работы и возможностей HASP HL API:

- HASP HL ToolBox – утилита с графическим интерфейсом, является частью пакета программ Vendor Center.
- Примеры API – для того чтобы изучить пример использования HASP HL API, выберите папку примера в соответствии с используемым вами компилятором.

### HASP HL ToolBox

Для применения HASP HL ToolBox вы должны обладать ключом HASP HL и соответствующим Кодом разработчика, чтобы получить к ключу доступ. Данная программа запускается из пакета Vendor Center. Для получения более подробной информации по работе с утилитой HASP HL ToolBox следует обратиться к ее справочной системе.

#### Доступ к API

HASP HL ToolBox представляет собой интерактивный интерфейс HASP HL API. Вы осуществляете вызовы API, которые затем переадресуются подключенному ключу HASP HL.

#### Возможности, предоставляемые HASP HL API

Программа HASP HL ToolBox позволяет:

- просмотреть исходный код, сгенерированный для вызова каждой функции, который в дальнейшем может быть скопирован и вставлен в исходный код приложения;
- оценить реализацию HASP HL API. Функции HASP HL API, включенные в HASP HL ToolBox, показываются на отдельном экране. Для осуществления вызова функции необходимо иметь специальную информацию, связанную с этой функцией;
- переносить буферы памяти на крипто-процессор ключа HASP HL. Программа также может использоваться для дешифрования буферов данных;
- использовать несколько собственных интерфейсов для работы с различными языками программирования и HASP HL API.

## Примеры API

Для ознакомления с процессом внедрения защиты HASP HL API в исходный код приложения вы можете просмотреть примеры приложений. Примеры содержат простые случаи работы API.

Среди установленных компонентов HASP HL имеются папки с различными интерфейсами и компиляторами. В каждой папке содержатся библиотеки API, файл заголовка и пример приложения. При использовании приложений-примеров к компьютеру следует подключить демонстрационный ключ HASP HL с маркировкой DEMOMA.



Информация о доступных примерах и определенных языках программирования доступна на интернет-сайте компании Aladdin и на установочном компакт-диске HASP HL.

## Реализация защиты с помощью HASP HL API

### Планирование требований

До применения HASP HL API необходимо принять ряд решений.

- а. Что вы хотите защитить? Вопрос может показаться очевидным, однако от ответа на него зависят места размещения вызовов HASP HL. Обычно предусматривается проверка наличия ключа при начале работы приложения. В дополнение к этому можно определить несколько аспектов работы вашего программного

обеспечения, которые вы хотели бы защитить с помощью вызовов HASP HL API.



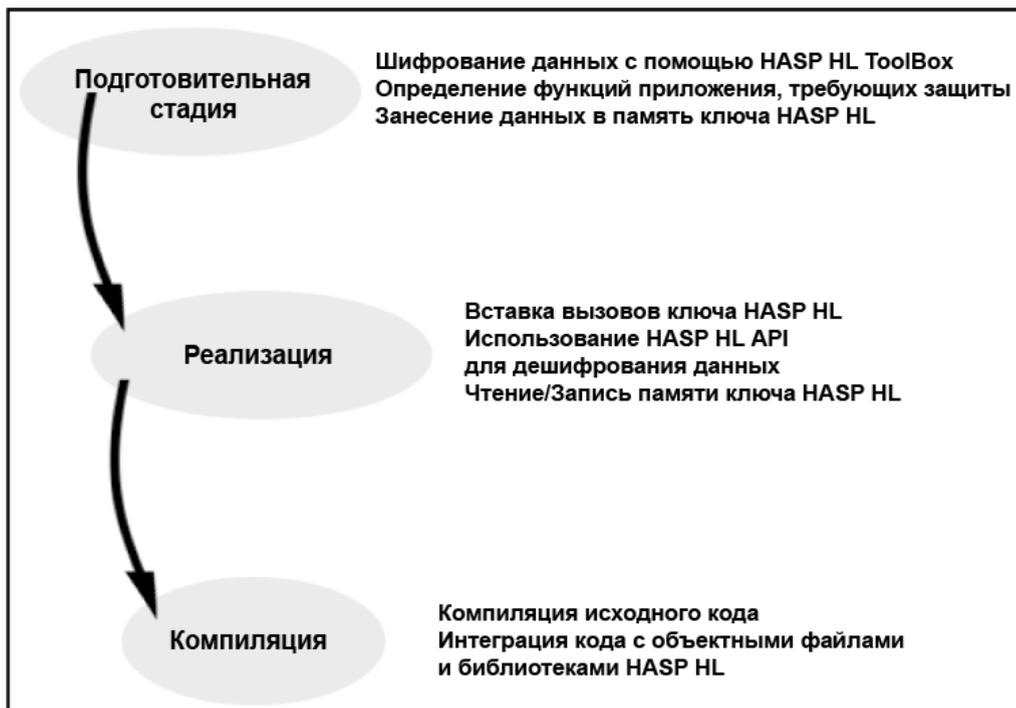
Защита отдельных функций осуществляется с помощью параметра **Feature ID**, которому необходимо присвоить номер. Для получения более подробной информации обратитесь к приложению С (описание `hasp_login`).

- b. Будут ли использоваться в моей схеме защиты зашифрованные данные? Если вы планируете их использовать, вам следует воспользоваться HASP HL ToolBox для шифрования выбранных данных. Затем зашифрованные данные вставляются в приложение при реализации защиты с помощью HASP HL API. Дешифрование будет производиться HASP HL во время работы приложения.
- c. Будут ли храниться данные в памяти ключа HASP HL? Если программное обеспечение защищается с помощью ключа HASP HL с памятью, конфиденциальные данные могут храниться непосредственно в памяти этого ключа. API позволяет использовать память ключа на чтение и запись. Для записи буферов данных в память ключа HASP HL следует использовать HASP HL ToolBox.

### **Стадии реализации защиты с помощью HASP HL API**

После осуществления описанных выше действий, можно переходить непосредственно к созданию защиты с помощью HASP HL API. Основные стадии этого процесса показаны на рисунке 5.1.

Рисунок 5.1



### Для реализации защиты с помощью HASP HL API

1. Изучите код примера, соответствующего вашей среде разработки.
2. Вставьте в исходный код приложения вызов ключа HASP HL для открытия сессии. У успешно открытой сессии будет собственный уникальный идентификатор.



Идентификатор сессии генерируется самостоятельно и применяется только к данной сессии подключения. Для получения более подробной информации обратитесь к приложению С (описание `hasp_login`).

3. После начала сессии для обмена данными с ключом могут использоваться и иные функции HASP HL API. Например, можно осуществить дешифрование важных данных, используемых приложением. Также можно считать данные, хранящиеся в памяти ключа HASP HL, установить метку времени и т.д.

4. По результатам предыдущего шага проверяются возможные несовпадения и для пользователя выводятся соответствующие уведомления.
5. Повторите шаги 2-4 по всему коду приложения.
6. Скомпилируйте исходный код.



После компиляции исходного кода в целях добавления дополнительного уровня защиты следует использовать HASP HL Envelope, что также предотвратит реинжиниринг защищенного кода.

## Функция Login в HASP HL API

Функция Login (открытие сессии) является ключевой в реализации защиты с помощью HASP HL API. Для обмена данными с ключом HASP HL необходима действующая сессия подключения. Действие функции Login в контексте HASP HL API показано на рисунке 5.2. Для того чтобы получить доступ к HASP HL, необходим Код Разработчика и идентификатор функции.

Прочие функции будут доступны только после осуществления успешного вызова на открытие сессии. Если HASP HL не обнаружен или не подключен к компьютеру, выводится сообщение об ошибке. Сообщение об ошибке также выводится в случае, если Код разработчика не соответствует подключенному ключу HASP HL.

**Рисунок 5.2** Схема выполнения вызова Login в системе HASP HL



---

## Возможности Login

Как и прочие аспекты реализации HASP HL API, вызовы на открытие сессии не зависят ни от платформы, на которой установлено защищенное приложение, ни от модели ключа HASP HL. Тем не менее, при задании вызова на открытие сессии, необходимо указать к чему вы пытаетесь получить доступ. Вы должны определить:

- Получаете ли вы доступ к особой или доступной по умолчанию функции.
- Каким образом будет происходить поиск ключа HASP HL.
- Включить или выключить связь с ключом HASP HL через сервер.

### Указание идентификатора функции (Feature ID)

Вы можете получить доступ, как к особой функции, так и к функции доступной по умолчанию. Функциям по умолчанию присваивается программный номер 0.

Идентификатор функции соответствует «PROGRAM NUMBER FEATURE» (Функции программного номера), существующей в ключе HASP HL. Особым функциям присваиваются программные номера, в то время как доступные по умолчанию функции остаются таковыми до тех пор, пока к компьютеру подключен ключ HASP HL. Определение программного номера особенно важно при лицензировании защищенного приложения. Более подробная информация по этому вопросу содержится в разделе приложения С, посвященном `hasp_login()`.

При получении доступа к лицензированной функции защищенное приложение проверяет не только наличие ключа HASP HL, но также и условия лицензии. Если лицензия имеет силу, доступ к функции предоставляется. Более подробная информация по этому вопросу приводится в разделе «Присвоение функциям идентификаторов».

Кроме того, существуют и прочие аспекты вызова Login, которые могут контролироваться при реализации защиты с помощью HASP HL API.

### Параметры поиска

По умолчанию защищенное приложение осуществляет поиск требуемого ключа HASP HL как на локальном компьютере, так и в сети. Параметры поиска могут быть сужены:

- Поиск ключа HASP HL только на локальном компьютере. Для задания этого ограничения следует включить в декларацию вызова `login` строку  
`HASP HL_PROGNUM_OPT_NO_REMOTE`
- Поиск ключа HASP HL только в сети. Для задания этого ограничения следует включить в декларацию вызова `login` строку  
`HASP HL_PROGNUM_OPT_NO_LOCAL`

### **Счетчик подключений к ключу**

В случае получения доступа к лицензии, хранящейся на ключе HASP HL Net, использование лицензии определяется методом подсчета количества подключенных рабочих станций. Однако это условие можно заменить на подсчет количества разрешений на запуск процесса. В результате счетчик лицензий будет уменьшаться с запуском каждого указанного процесса. Для задания такого статуса следует включить в декларацию вызова login строку  
HASP\_HL\_PROGNUM\_OPT\_NO\_PROCESS

### **Обнаружение терминального сервера**

По умолчанию HASP HL обнаруживает и прекращает работу защищенных приложений на удаленном терминальном сервере. Для изменения значения данного параметра на обратное следует включить в декларацию вызова login строку  
HASP\_HL\_PROGNUM\_OPT\_TS

### **Разрешить доступ к ключам HASP3/HASP4**

По умолчанию HASP HL не дает доступ к более ранним версиям ключей HASP. Для преодоления этого ограничения следует включить в декларацию вызова login строку

HASP\_HL\_PROGNUM\_OPT\_CLASSIC

После получения доступа к ключу HASP HL и начала сессии в ваше распоряжение попадает широкий спектр функций HASP HL API, которые можно использовать для построения надежной схемы защиты. Следующий раздел описывает эти возможности.



Каждая сессия, начатая после успешного вызова login, должна заканчиваться соответствующим вызовом logout. Функции шифрования и дешифрования с помощью алгоритма AES будут недоступны при использовании ключей, не являющихся HASP HL.

## Доступные функции HASP HL API

Степень защищенности приложений зависит от того, как реализуется схема защиты с помощью HASP HL API. Включенные в код вызовы аппаратного ключа полностью контролируют доступ к приложению.

В данном разделе рассматриваются возможности HASP HL API, доступные после начала сессии. Вопросы оптимизации разработки защиты с помощью HASP HL API рассматриваются в разделе «Оптимальная реализация защиты с помощью HASP HL API». Для ознакомления с работой HASP HL API следует использовать HASP HL ToolBox. Все упомянутые ниже функции подробно рассматриваются в приложении «Справочник по HASP HL API».

### Группы функций

Доступные функции HASP HL API можно условно разделить на пять групп:

- сессионные функции;
- функции памяти;
- функции шифрования;
- функции времени;
- функции управления.

### Сессионные функции

Для выполнения большинства функций HASP HL API необходимо предварительно осуществить вызов login и получить, таким образом, доступ к лицензии, хранящейся на определенном аппаратном ключе. Доступные возможности описываются в разделе «Возможности Login». Каждый вызов login должен сопровождаться по окончании сеанса вызовом logout. Между этими двумя вызовами могут осуществляться другие вызовы ключа HASP HL.

### Функции памяти

HASP HL API может применяться для:

- чтения буферов данных, хранящихся в памяти ключа HASP HL;
- записи буферов данных в память ключа HASP HL.

Размер буфера данных ограничивается размером памяти используемой модели ключа. Размер памяти различных моделей аппаратных ключей приведен в таблице D.2 «Технические спецификации моделей ключей HASP HL».



Используйте память аппаратного ключа для хранения информации, которая может использоваться для проверки и идентификации конечного пользователя.

Задействуйте память для хранения данных, используемых приложением во время его работы. Доступ к этой конфиденциальной информации будет являться частью общей схемы защиты.

## Функции шифрования

После успешного начала сессии с ключом HASP HL вы можете осуществлять шифрование и дешифрование буфера данных с помощью встроенного крипто-процессора, использующего симметричное шифрование. Симметричное шифрование предусматривает использование одного ключа как для шифрования, так и для дешифрования.

## Функции времени

Если вы используете ключ HASP HL Time, API может применяться для доступа к часам такого ключа. Это позволяет:

- устанавливать время;
- считывать время.

В HASP HL API также входят две функции обработки даты и времени.

## Функции управления

В API входят функции, позволяющие получать информацию о текущей сессии подключения к ключу, статусе ключа, обновлять лицензии.

### Обновления

HASP HL API, с помощью специального вызова, может применяться для создания обновлений. В отличие от остальных функций API, обновление не требует вызова на начало сессии. Более подробная информация содержится в разделе приложения C, описывающем вызов `hasp_update()`. Эта функция лежит в основе системы дистанционного перепрограммирования.

## HASP ID

Вы можете указать серийный номер ключа HASP HL — HASP ID — что позволит привязать такие действия как обновление лицензии или получение даты с ключа, к определенному ключу HASP HL.

# Защита с помощью HASP HL Envelope

В главе рассматриваются следующие темы, описывающие защиту приложений с помощью HASP HL Envelope:

- условия применения HASP HL Envelope;
- работа с HASP HL Envelope;
- функции создания защиты с помощью HASP HL Envelope;
- работа с файлами данных в HASP HL Envelope.

## Возможности HASP HL Envelope

HASP HL Envelope создает вокруг приложения защитную оболочку. Данный инструмент предоставляет в ваше распоряжение несколько мощных защитных функций, повышающих общий уровень защищенности вашего программного обеспечения.

Рассматриваемый инструмент применяется для защиты исполняемых файлов и библиотек DLL. HASP HL Envelope также содержит антиотладочные и антиреинжиниринговые средства.

Применение HASP HL Envelope при создании защиты вашего программного обеспечения позволяет создать связь между ним и аппаратным ключом HASP HL. В случае если эта связь нарушается (то есть, защищенное программное обеспечение не может получить доступа к соответствующему ключу), программа перестает работать до тех пор, пока не будет подключен нужный ключ.

Построение защиты с помощью HASP HL Envelope является наиболее быстрым способом обеспечения безопасности программы без изменения ее исходного кода. Работа с HASP HL Envelope осуществляется с помощью графического интерфейса, который позволяет выполнять следующие действия:

- Защищать программы.
- Задавать параметры защиты.
- Указывать Код разработчика для аутентификации наличия определенного ключа HASP HL.
- Редактировать сообщения, выводимые на экран компьютера, во время работы защищенного приложения.

В дополнение к созданию связи между защищенным приложением и аппаратным ключом, HASP HL Envelope создает вокруг указанного файла приложения многочисленные защитные слои.



Создание случайных защитных слоев вокруг защищаемого приложения гарантирует, что реализуемые стратегии защиты для различных приложений отличаются.

---

## Условия применения HASP HL Envelope

Для применения HASP HL Envelope необходимо, чтобы указанное ниже программное обеспечение было установлено.

### Требования к программному обеспечению

Все указанные компоненты должны быть установлены на вашей системе:

- Драйверы или демоны HASP HL.
- Программное обеспечение HASP HL, включая комплект утилит Vendor Center.
- Необходимый Код Разработчика должен находиться в папке Vendorcodes (см. раздел «Получение Кода Разработчика с мастер-ключа HASP HL»).
- Одна из двух операционных систем – Windows 2000 или Windows XP.
- Исполняемые файлы, которые вы хотите защитить.

## Работа с HASP HL Envelope

Для запуска Vendor Center выберите в главном меню Windows:

Start/Programs/HASP HL/Vendor Center.

После запуска Vendor Center можно открыть **HASP HL Envelope**.

### Основная процедура защиты

Приведенная ниже процедура описывает процесс создания защиты приложения с помощью HASP HL Envelope. Для получения более подробной информации по этой процедуре следует обратиться к справочной системе HASP HL Envelope.

1. Запустите **HASP HL Envelope** из оболочки Vendor Center.
2. Выберите программу или программы, которые вы хотите защитить.
3. Определите параметры защиты программы или программ.
4. Осуществите защиту программы (программ).
5. Защищенное программное обеспечение вместе с ключами HASP HL может поставляться клиентам.

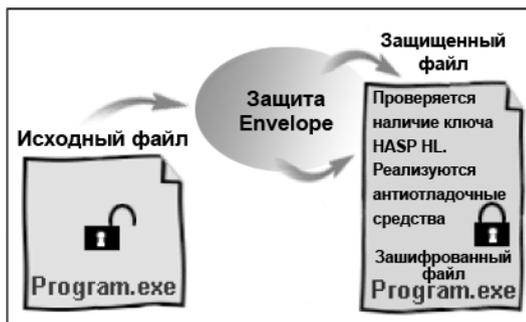


HASP HL Envelope не вносит изменения в защищаемые файлы. Тем не менее, для защищенного приложения следует отвести отдельную папку в целях разделения исходного и защищенного файлов.

HASP HL Envelope применяет заданные параметры защиты к незащищенному коду. HASP HL Envelope не оказывает влияния на работу защищенного приложения. Единственное изменение заключается в необходимости наличия подключенного к компьютеру определенного ключа HASP HL. Если ключ подключен, защищенный файл можно использовать.

Логика защиты с помощью HASP HL Envelope показана на рисунке 6.1.

**Рисунок 6.1** Работа HASP HL Envelope



## Работа с HASP HL Envelope в режиме командной строки

HASP HL Envelope можно запустить из командной строки. Такая возможность особенно полезна, когда необходимо запустить автоматический процесс, не требующий графического интерфейса. Версия HASP HL Envelope для командной строки запускается с открытием файла `envelope.com`, находящегося в папке:

*Program Files/Aladdin/HASP HL/Vendortools/Vendorcenter*



Версия программы для командной строки используется преимущественно для автоматических процессов. Перед ее запуском создайте и сохраните проекты защиты, а также конфигурационный файл с помощью *envelope.exe*.

### Запуск версии для командной строки

В командной строке напечатайте `ENVELOPE`. Система осуществляет поиск и запуск приложения *envelope.com*.

### Параметры для командной строки

С версией программы для командной строки можно использовать следующие параметры:

Таблица 6.1 Параметры командной строки для HASP HL Envelope

Команда	Описание
-h /--help	Выводит список возможных параметров для запуска программы из командной строки. Чтобы вернуться к командной строке, нажмите <b>Enter</b> .
-p/--protect <конфигурация>	Указанный конфигурационный файл используется как входные данные для создания вокруг приложения защитной оболочки. Конфигурационные файлы содержат определения параметров защиты. При использовании файла конфигурации защищаться может только один файл.
-p/--protect <проект>	Программа будет воспринимать указанный проект как исходные данные, которые необходимо защитить. Операции по созданию защиты будут подвергнуты все файлы проекта.
<проект>	Запускается версия с графическим интерфейсом с указанным проектом в качестве текущего.

## Параметры защиты с помощью HASP HL Envelope

В данном разделе рассматриваются возможные установки параметров для создания защиты.

HASP HL Envelope является методом автоматической защиты вашего приложения. Тем не менее, вы можете самостоятельно определить и настроить параметры будущей защиты программного обеспечения. Настройки разделены на две категории: обязательные параметры и предустановленные настройки, оказывающие влияние на поведение защищенного файла.

### Обязательные параметры

Для создания защиты с помощью HASP HL Envelope необходимо указать:

- путь к исходному файлу — чтобы защитить приложение, утилита HASP HL Envelope должна знать, где оно находится;
- Код разработчика — его необходимо указать, чтобы получить доступ к подключенному ключу HASP HL. При первом запуске HASP HL Envelope Кодом разработчика по умолчанию является DEMOMA.

Это достаточная информация для начала процесса защиты приложения. Все остальные параметры защиты используют значения по умолчанию, приведенные ниже.

### Дополнительные и предустановленные настройки

Дополнительные настройки появляются на трех отдельных экранах HASP HL Envelope. Значение любого параметра, заданного по умолчанию, может быть изменено. Эти настройки влияют на свойства и поведение защищенного приложения. Все настройки подробно описываются в справочной системе HASP HL Envelope.

### Поиск ключа HASP HL

HASP HL Envelope позволяет задать способ поиска приложением ключа HASP HL. Возможны следующие варианты:

- на локальном компьютере — защищенное приложение осуществляет поиск ключа только на локальном компьютере;
- В сети - защищенное приложение осуществляет поиск ключа только в сети;
- на локальном компьютере и в сети — защищенное приложение в первую очередь осуществляет поиск ключа на локальном компьютере, а затем в сети (по умолчанию).

## Поведение защищенной программы

HASP HL Envelope позволяет задать и другие свойства защищенного приложения:

- Как часто на ключ посылаются запросы. При выполнении запросов используются операции шифрования и дешифрования.
- Временной интервал между проверками на присутствие требуемого ключа HASP HL.
- Включение и выключение поддержки программ, требующих оверлеев.
- В случае применения схем лицензирования, предусматривающих одновременное использование приложения, HASP HL Envelope позволяет проводить подсчет либо количества процессов, либо количества станций (см. также раздел «HASP License Manager»)
- Время ожидания приложением окончания загрузки драйверов устройств HASP HL.

## Свойства защиты

Для защищенного приложения можно задать определенные параметры безопасности. HASP HL Envelope может настраиваться по следующим параметрам:

- Обнаружение системных и пользовательских отладочных средств. Можно активировать средства HASP HL, которые будут блокировать возможные попытки раскрытия схемы защиты.
- Определение количества слоев защитных модулей, создаваемых вокруг приложения. Диапазон возможных значений — 1-50. Значение по умолчанию — 12.



Увеличение количества защитных модулей увеличивает время загрузки приложения и размер файла. Также существует зависимость между уровнем шифрования, размером файла и скоростью запуска — чем выше уровень шифрования, тем больше размер файла и дольше время загрузки приложения.

- Можно указать частоту обращений к ключу HASP HL для осуществления шифрования. Данная настройка контролирует компактность вызовов ключа защищенным приложением. Ползунок **Encryption Level (Уровень шифрования)** позволяет задать частоту этих обращений.

**Рисунок 6.2** Ползунок Encryption Level



### Работа с файлами данных

Во время работы вашему защищенному приложению может понадобиться доступ к файлам данных. HASP HL Envelope позволяет контролировать доступ к файлам данных с помощью следующего механизма контроля:

- Фильтры данных. Вы можете определить типы файлов, к которым во время работы может обращаться защищенное приложение. Также можно определить какие файлы будут исключены из этого списка.
- Ключи шифрования данных. Ключ шифрования состоит из восьми символов; ключ используется для шифрования и дешифрования данных.



В случае, когда несколько приложений получают доступ к одному набору документов, следует использовать один ключ шифрования.

Фильтры данных и ключ шифрования являются частью схемы защиты приложения. Шифрование файлов данных осуществляется с помощью приложения *dfcrypt.exe* (командная строка). Более подробная информация по вопросу шифрования также содержится в разделе «Шифрование файлов данных».

### Поддержка работы пользователя

Вы можете задать различные сообщения, выводимые на экран при работе пользователя с защищенным с помощью HASP HL Envelope приложением. У HASP HL Envelope есть набор кодов сообщений. Каждый код связан с соответствующим сообщением, которое выводится на экран при работе с защищенным приложением. Выводимые сообщения могут быть изменены.

Кроме того, вы можете предусмотреть вывод на экран специального сообщения при запуске защищенного приложения. В сообщении можно, например, проинформировать пользователя о возможных задержках в работе приложения из-за необходимости дешифрования требуемых данных.

## HASP HL Envelope для приложений Mac

С помощью HASP HL Envelope можно осуществлять защиту Mac-приложений. HASP HL Envelope для Mac представляет собой терминальную утилиту OS X, позволяющую защищать приложения Mach-O.

### Защита приложений Mac

Приложения Mac защищаются путем:

- a. Определения характеристик хранения параметров защиты в файле конфигурации.
- b. Создания копии связки файлов, содержащей исполняемый компонент Mach-O.
- c. Получения доступа к файлу конфигурации в течение сеанса защиты.

Параметры настройки HASP HL Envelope подробно описаны в файле *Envelope Configuration Settings.pdf*, который находится в папке *Doc* установочного компакт-диска HASP HL.

После того, как параметры защиты были определены и занесены в файл конфигурации, вы можете активировать Envelope следующим образом:

```
hasphlmacho_Darwin -c configuration.cfg
```

### Переназначение установок файла конфигурации

Несколько указанных в файле конфигурации настроек можно переназначить с помощью дополнительных аргументов, указанных в таблице 6.2.

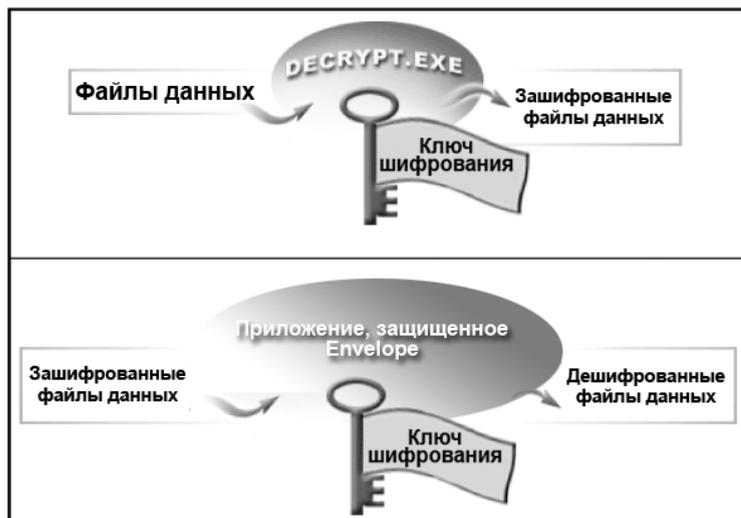
**Таблица 6.2** Установки конфигурационного файла HASP HL Envelope

Установка	Описание
-c <имя файла конфигурации>	Обязательная установка имени файла конфигурации.
-i <имя исходного файла>	Дополнительная установка, указывающая исходный исполняемый компонент Mach-O в bundle файла приложения.
-o <имя получаемого в результате файла>	Дополнительная установка, указывающая получаемый в результате исполняемый компонент Mach-O в связке файла приложения.
-v <название файла с кодом разработчика>	Дополнительная установка, указывающая код разработчика, используемый защищенным файлом для доступа к ключу HASP HL.
-m <имя файла сообщений>	Дополнительная установка, указывающая файл, в котором содержатся сообщения, выводимые на экран во время работы приложения.

## Шифрование файлов данных

Защищенные с помощью HASP HL Envelope приложения могут получать доступ к зашифрованным файлам данных, определяемых фильтрами данных и ключом шифрования. HASP HL Envelope не шифрует файлы данных. Для шифрования используется приложение *dfcrypt.exe*. Оно генерирует файлы данных, обрабатываемые в дальнейшем исполняемыми файлами, защищенными с помощью HASP HL Envelope.

**Рисунок 6.3** Работа с файлом данных с помощью *dfcrypt.exe*



### Применение *dfcrypt.exe*

Данный инструмент имеет интерфейс командной строки. Для нормальной работы ему необходимо задать особый набор параметров. После того, как эти параметры будут применены к заданному набору файлов данных, HASP HL Envelope сможет получить доступ к зашифрованным файлам. На Рисунке 6.3. показана процедура работы с файлами с помощью *dfcrypt.exe*. Зашифрованные этой программой файлы данных могут затем дешифроваться и использоваться приложением, защищенным с помощью HASP HL Envelope. Однако это будет возможно только в том случае, если в процессе создания защиты вы зададите ключ шифрования в HASP HL Envelope.

## Инструкции по применению *dfcrypt.exe*

Для применения *dfcrypt.exe* следует определить:

- команду;
- список исходных файлов и директорий;
- место размещения зашифрованных данных.

При указании нескольких исходных файлов и директорий следует указывать уже существующую директорию.

Для ввода параметров следует использовать следующий формат:

```
Dfcrypt <опция> исходные данные конечные данные
```

Например:

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

## Доступные команды *dfcrypt.exe*

В таблице 6.3. приводится перечень доступных команд *dfcrypt*. Знак звездочки показывает обязательные команды.

**Таблица 6.3** Перечень возможных команд *dfcrypt.exe*

Команда	Действие
-e, --encrypt	Шифрует данные, доступна по умолчанию.
-d, --decrypt	Дешифрует данные
-c, --vcf:<файл>	Указывает файл кода разработчика HASP HL*
-k, --key:<ключ>	Указывает ключ для применения при шифровании файла данных. Должен содержать 8 печатаемых символов.*
-o, --overwrite	Записывает поверх существующего файла.
-r, --recursive	Позволяет включать в операцию поддиректории.
-h, --help	Показывает перечень команд <i>dfcrypt</i> .
-q, --quiet	Исключает информацию о правах и показ индикатора осуществления действия. Показываются только сообщения об ошибках. Команда полезна при интеграции Makefile.



# Стратегии защиты

В главе рассматриваются следующие темы, описывающие стратегии улучшения защиты, реализуемой с помощью HASP HL:

- наилучшие подходы к реализации защиты с помощью HASP HL;
- оптимизация защиты, реализуемой с помощью HASP HL API;
- противодействие атакам с помощью оптимальных техник защиты приложений;
- рекомендации по повышению уровня безопасности защищенного приложения.

Система HASP HL является на сегодняшний день наилучшим программно-аппаратным комплексом по защите программного обеспечения. Степень получаемой защиты зависит от реализуемой схемы, которая должна наилучшим образом соответствует вашим требованиям.



Для достижения максимальной степени защиты с помощью HASP HL следует использовать и Envelope, и API.

Если вам необходима дополнительная помощь в усилении защиты программного обеспечения, мы можем предложить поддержку команды наших консультантов. Они могут помочь вам в решении широкого спектра вопросов — включая стратегии защиты и техники реализации защиты.

Свяжитесь с местным представительством Aladdin, если вам нужна информация об оказываемых консультациях.

## Защита с помощью HASP HL – наилучшие подходы

Два метода реализации защиты с помощью HASP HL уже были представлены в предыдущих главах. Каждый метод защиты имеет свои преимущества, а у разных пользователей требования отличаются.

Для принятия решения о том, какой метод подходит вам лучше, мы рекомендуем вам обратиться к разделу «HASP HL API в сравнении с HASP HL Envelope».

При реализации защиты программного обеспечения рекомендуется следовать следующим общим рекомендациям:

- a. Используйте для защиты программного обеспечения и HASP HL Envelope, и HASP HL API.
- b. Избегайте излишне упрощенных стратегий защиты.

### Используйте оба метода защиты с помощью HASP HL

Применение обоих методов – и HASP HL Envelope, и HASP HL API – позволяет существенно повысить защищенность программного обеспечения. Постройте защиту на основе вызовов ключа с помощью HASP HL API, а после компиляции кода защитите исполняемый файл с помощью HASP HL Envelope.



После компиляции исходного кода, с помощью HASP HL Envelope создайте вокруг ваших исполняемых файлов и библиотек защитную оболочку.

В последующих разделах рассматривается оптимизация каждого метода. Однако имейте в виду, что использование одного метода не исключает использование другого. Это особенно касается тех, кто применяет HASP HL API. Для создания защиты исполняемого файла с помощью HASP HL Envelope требуется менее пяти минут!

### Изменяйте схему защиты

Для достижения высокого уровня безопасности программного обеспечения следует часто менять используемые схемы защиты. Меняйте используемые методы, реализуя новые средства обеспечения безопасности в каждой версии вашего приложения. Регулярно обновляйте используемые для создания защиты инструменты.

Aladdin постоянно совершенствует собственную линейку продуктов. Мы рекомендуем время от времени посещать сайт Aladdin в целях получения информации о новых возможностях защиты с помощью системы HASP HL. В комплекте утилит Vendor Center есть ссылка на сайт Aladdin. Используйте ее, чтобы получать последние обновления программного обеспечения.

Для получения последних обновлений вы также можете связаться с ближайшим представителем Aladdin. Поддерживайте постоянный контакт с Aladdin и всегда будьте на один шаг впереди тех, кто пытается атаковать ваше программное обеспечение.

## Оптимизация защиты с помощью HASP HL API

В данном разделе описываются аспекты защиты программного обеспечения, на которые вам необходимо обратить внимание, а также приводятся советы и стратегии защиты. Принимая во внимание то, что аппаратные средства HASP HL практически невозможно взломать или воспроизвести, возможные атаки, скорее всего, будут сосредоточены на программном обеспечении. Взломщики будут стараться осуществить трассирование защищенного кода и устранили схемы защиты.

### Аспекты защиты программного обеспечения

Большинство издателей программного обеспечения беспокоятся о двух наиболее распространенных методах атак на защищенное программное обеспечение:

- эмулирование вызовов защитного устройства;
- эмулирование программного обеспечения, используемого производителем аппаратного средства защиты.

Для эмулирования вызовов защитных процедур незаконный пользователь должен таким образом изменить исполняемый файл, чтобы он не посылал запросов на ключ, не проверял возвращаемые ключом результаты или не осуществлял действий в соответствии с результатами, заданными в коде приложения. Такой вид атаки используется в случае, если предполагается низкий уровень реализации защиты.

Для эмуляции программного обеспечения производителя ключа, осуществляющее атаку лицо обычно пытается манипулировать вызовами, применяемыми для обмена данными с аппаратным ключом. Такой подход позволяет достичь того, что ключ выдает «верные» ответы, даже если требуемое устройство HASP HL не подключено.

Необходимо заметить, что применение этих методов различается в зависимости от приложения, то есть они не могут использоваться в том же виде для атаки на другие защищенные приложения.



Для достижения максимального уровня безопасности мы рекомендуем устанавливать связь вашего приложения с объектами HASP HL, а не библиотеками DLL. Это улучшает защиту, поскольку объекты компилируются как неотъемлемая часть получаемого в результате исполняемого файла, что затрудняет процедуру взлома.

## Оптимальная реализация защиты с помощью HASP HL API

В данном разделе приводятся советы и приемы борьбы с ранее описанными атаками. Максимизировать уровень защиты вашего приложения позволит использование как можно большего числа рассмотренных приемов.

Помните, что, строя защиту, нельзя забывать об обычном пользователе защищенного приложения. Например, простой пользователь может забыть подключить ключ к компьютеру и реализуемая вами защита должна предусматривать такую возможность. «Умные» стратегии защиты должны бороться с атаками взломщиков, а не законными пользователями.

Данные рекомендации помогут улучшить защиту с помощью HASP HL API:

- необходимо предусмотреть большое число вызовов ключа HASP HL;
- используйте возможности ключа для шифрования и дешифрования данных;
- используйте сложные вызовы HASP HL API;
- используйте память ключа HASP HL

## Используйте большое число вызовов HASP HL API

Внедрение большого количества вызовов ключа истощит силы пытающихся взломать ваше программное обеспечение. Комплексная реализация HASP HL API, включающая множественные вызовы аппаратного ключа, повышает сложность трассировки ваших схем защиты.

Чем больше вызовов и возвращаемых кодов, проверяемых системой HASP HL, тем больше сложностей с трассировкой и удалением каждого из них. Вызовы должны осуществляться из максимально возможного количества мест программного кода.

## Осуществляйте шифрование/дешифрование данных с помощью ключа HASP HL

Не останавливайтесь на простом увеличении количества вызовов, включенных в программный код — усложняйте сами вызовы. Вызов открытия сессии, проверяющий присутствие ключа HASP HL, представляет собой самый простой элемент защиты. Используйте возможности

ключа таким образом, чтобы затруднить неавторизованный доступ к программному обеспечению.

Шифрование данных с помощью встроенного в ключ крипто-процессора значительно повышает защищенность приложения. Шифрование данных требует не просто присутствия ключа, но также и задействует его криптографические возможности.

Реализация защиты с помощью HASP HL API, при которой данные дешифруются определенным ключом, создает между аппаратным ключом и защищенным приложением связь, которую трудно удалить. В этом случае необходимо не только взломать приложение, но также и дешифровать данные.



Процессы шифрования и дешифрования происходят в самом ключе HASP HL, что лежит за пределами возможностей антиотладочных средств.

Нет необходимости в шифровании всех данных защищенного приложения; вместо этого следует сосредоточиться на ключевых данных — заголовки файлов, важные постоянные, необходимые для расчетов, определенные поля баз данных. Необходимо рассмотреть возможность шифрования тех данных, которые влияют на выполнение основных функций приложения.

## Процесс шифрования

Процесс шифрования простых наборов данных приводится ниже. Эти простые процедуры могут изменяться в зависимости от ваших требований.

1. Подключите ваш ключ HASP HL к компьютеру.



Убедитесь в том, что в процессе обмена данными будет использоваться верный код разработчика!

2. Зашифруйте данные.

Для этого следует использовать HASP HL ToolBox. Для получения более подробной информации по шифрованию обратитесь к справочной системе самой программы.

3. Вставьте полученный код в исходный код вашего приложения. Замените исходные понятные данные зашифрованными. Теперь вы можете дешифровать данные в процессе работы приложения.
4. Используйте функцию `hasp_decrypt()`, более подробно описанную в приложениях, для дешифрования зашифрованной информации с помощью ключа HASP HL.  
Теперь с данными могут осуществляться различные операции. Не забудьте предусмотреть вывод сообщения об ошибке в случае, если соответствующий ключ HASP HL не подключен к компьютеру.



Не следует шифровать сообщения, схожие с «HASP HL not found» («Ключ HASP HL не найден»). Подобные сообщения выводятся на экран в том случае, когда ключ HASP HL не подключен к компьютеру; их корректное дешифрование не должно зависеть от наличия ключа HASP HL.

### Избегайте повторяющихся схем

Если вы интегрируете вызовы API в исходный код приложения схожим образом, то схему защиты будет легче распознать, отследить и взломать. Если взломщик знает что искать, защита будет под угрозой.

Используйте широкий спектр предлагаемых HASP HL API функций, но постарайтесь избежать повторяющихся приемов.



После того, как исходный код будет скомпилирован, наилучшим способом повысить защищенность вашего приложения будет являться использование HASP HL Envelope для создания защитной оболочки вокруг исполняемых файлов и библиотек вашего приложения. Используйте различные схемы защиты для каждого исполняемого файла.

### Разделите процедуру проверки наличия ключа HASP HL

Проверка наличия HASP HL проходит в три шага:

- a. с помощью Кода разработчика и иных параметров осуществляется открытие сессии с ключом HASP HL;
- b. оцениваются возвращенные значения;
- c. осуществляются действия, соответствующие возвращенным значениям.

Не следует реализовывать описанный выше процесс последовательно. Скорее, лучше разнести его по программному коду приложения. Разде-

ленные шаги намного тяжелее отследить, чем последовательные. Также следует отсрочить реакцию приложения на отсутствие ключа HASP HL.

Например, проверка наличия ключа HASP HL может осуществляться, когда пользователь выбирает определенный пункт меню. Позвольте пользователю продолжать работать, даже если ключ не подключен к компьютеру. Предусмотрите вывод сообщения об отсутствии ключа после выполнения совершенно иной операции. В этом случае трудно увидеть связь между какой-то определенной функцией и проверкой наличия ключа HASP HL.

### **Используйте ключ HASP HL с памятью**

Использование ключа HASP HL с памятью существенно расширяет возможности защиты приложения с помощью HASP HL API. В памяти ключа можно хранить значения, необходимые для корректной работы приложения.

Функции HASP HL API следует использовать для того, чтобы ваша программа считывала из памяти ключа данные, не проверяя их. Только «корректные» значения позволят программе работать правильно. Трудно предположить, что кому-то удастся верно угадать требуемое значение, позволяющее программе корректно функционировать.

Можно использовать память ключа HASP HL в целях предотвращения ситуации, когда один ключ позволяет загружать приложение на разных терминалах. Используйте HASP HL API для генерирования и хранения значения в памяти ключа HASP HL и в оперативной памяти при запуске приложения. Периодически осуществляйте проверку равенства этих двух значений.

### **Работа с кодом приложения и HASP HL API**

В разделе описываются приемы работы с кодом приложения и реализации защиты с помощью HASP HL API. Советы носят общий характер и могут применяться ко всем потенциальным пользователям системы HASP HL.

- Маскируйте вызовы API.
- Проверяйте контрольные суммы кода.
- Используйте данные, хранящиеся на ключе HASP HL.

### **Создавайте шумы, маскируя вызовы API**

Отвлекайте внимание взломщика с помощью шифрования данных, не оказывающих особого влияния на работу приложения. Такие «шумовые» вызовы API вводят атакующего в заблуждение и являются дополнительным препятствием при попытке трассировки и взлома вашей схемы защиты.

Вы можете генерировать шумы с помощью генератора случайных чисел, значений времени, промежуточных результатов вычислений и пр. Разумеется, такие вызовы не должны приводить к каким-либо результатам или действиям.

## Проверка контрольной суммы кода

Проверка контрольной суммы позволяет выявить факт изменения кода защищенного приложения.

Для осуществления простой проверки:

1. Определите контрольную сумму.
2. Сравните полученный результат с верным значением. Обязательно разделите шаги, как это было показано в разделе «Разделите процедуру проверки наличия ключа HASP HL»
3. Включите вывод сообщения об ошибке, в случае если значения не сошлись.

К сожалению, этот метод уязвим:

- Код может быть изменен таким образом, чтобы проверка контрольной суммы не осуществлялась.
- Можно вставить возврат верного значения контрольной суммы.

От подобных атак можно защититься, осуществляя операцию XOR. Можно также использовать CRC или иной алгоритм, чувствительный к последовательности байтов.

Другой подход заключается в исключении сравнения полученной суммы и заранее вычисленного результата. Вместо этого можно использовать контрольную сумму для выполнения действия, которое приводит к ошибке, если было вычислено неверное значение.

Например, храните результат контрольной суммы в качестве переменной, а в дальнейшем используйте его в качестве ключа при дешифровании кода или данных. Такой подход имеет преимущество отсроченной реакции. Кроме того, значение контрольной суммы не хранится в приложении в явном виде.

## Используйте различные возможности в качестве реакции на отсутствие ключа

Существует широкий выбор возможных ответов на отсутствие аппаратного ключа. Обычная реакция — вывод на экран сообщения «Ключ HASP HL не найден». Однако, это дает ясно понять, что была осуществлена проверка наличия ключа.

Необходимо рассмотреть возможности нарушения нормальной работы программы в случае отсутствия ключа. Неавторизованные пользователи

могут отнести это на счет ошибок работы программы. Они не должны понять, что была проведена проверка наличия ключа, а возникающие проблемы - ее непосредственный результат.

Например, можно сделать так, что приложение не будет отвечать на действия пользователя (например, на щелчки мыши), если верный ключ HASP HL не подключен к компьютеру. Другие программы при этом продолжают функционировать нормально.

При реализации защиты не забывайте, что простой пользователь, может просто забыть подключить свой ключ.

### **Используйте данные, хранящиеся в памяти ключа HASP HL**

При использовании данных, хранящихся в памяти ключа HASP HL, вы обычно проверяете верность получаемого с ключа значения. Однако процесс проверки заставляет включать реальное значение в само приложение. Следовательно, значение становится уязвимым.

Не следует явным образом проверять значение данных. Вместо этого следует:

1. Читать данные, хранящиеся в памяти ключа HASP HL.
2. Рассчитать контрольную сумму (см. раздел «Расчет контрольной суммы кода» выше).
3. Проверить контрольную сумму.

Эта процедура позволяет получить доступ к данным, хранящимся в памяти ключа HASP HL, без явной проверки их верности. В противном случае значение считается неверным и приведет к ошибке.

## **Наилучшие способы применения HASP HL Envelope**

При применении HASP HL Envelope необходимо сделать выбор между уровнем защиты и:

- размером защищаемого приложения;
- требуемым временем запуска приложения.

При установке высокого уровня защиты увеличится размер защищаемого приложения, также увеличится время загрузки.

Если размер файла и время загрузки приложения не имеют значения, можно установить самый высокий уровень защиты.

Значение, установленное в HASP HL Envelope по умолчанию, дает оптимальный баланс этих факторов. HASP HL Envelope самостоятельно создает эффективную защитную оболочку приложения, оптимизируя размер защищенного файла и время загрузки.

## **Используйте фоновые проверки**

Рекомендуется использовать фоновые проверки присутствия аппаратного ключа. Если пользователь отключает ключ HASP HL, чтобы загрузить приложение на другом терминале, защищенное приложение попросит пользователя вновь подключить ключ. В противном случае приложение не будет работать (см. также раздел «Свойства защиты» и справочную систему HASP HL Envelope).

## **Обновляйте программное обеспечение HASP HL**

Вам следует периодически обновлять программное обеспечение HASP HL. Обновления можно загрузить с сайта Aladdin.

Для загрузки последних версий программного обеспечения HASP HL можно воспользоваться ссылкой из набора утилит Vendor Center.

Компания Aladdin Knowledge Systems постоянно совершенствует технологию защиты программного обеспечения, следя, чтобы оно соответствовало всем современным требованиям. Обновления программного обеспечения HASP HL выходят регулярно.

Для адекватной защиты вашей системы следует использовать самые современные технологии.

# Обзор лицензирования с помощью HASP HL

В главе рассматриваются следующие темы, касающиеся лицензирования с помощью HASP HL:

- ключевые концепции лицензирования с помощью HASP HL;
- доступное программное и аппаратное обеспечение HASP HL, предназначенное для лицензирования;
- планирование схем лицензирования с применением HASP HL;
- работа с лицензиями с помощью HASP HL.

В следующих двух разделах подробно рассматривается программное обеспечение, предназначенное для осуществления лицензирования с помощью HASP HL.

Подробная информация по HASP HL Factory содержится в разделе «Лицензирование с помощью HASP HL Factory».

Подробная информация по Системе дистанционного перепрограммирования содержится в разделе «Система дистанционного перепрограммирования».

## Ключевые концепции лицензирования с помощью HASP HL

В целях максимально эффективного использования возможностей лицензирования системы HASP HL вам следует ознакомиться основными понятиями, связанными с процессом лицензирования. После того, как вы ознакомитесь с концепциями, вы сможете разработать план применения системы HASP HL для лицензирования вашего приложения.

В разделе рассматриваются следующие концепции:

- автоматическая система лицензирования;

- функция;
- лицензия;
- обновление;
- одновременное использование.

### **Автоматическая система лицензирования**

Эта концепция первый раз рассматривалась в данном документе в разделе Автоматическая система лицензирования. С точки зрения лицензирования, необходимо еще раз подчеркнуть несколько ключевых моментов:

- а. В системе HASP HL лицензирование и защита рассматриваются как два отдельных процесса. Вы защищаете программное обеспечение с помощью одного или обоих методов защиты HASP HL:
  - HASP HL API (см. раздел «Защита с помощью API системы HASP HL»)
  - HASP HL Envelope (см. Раздел «Защита с помощью HASP HL Envelope»)



С точки зрения лицензирования, не имеет значения то, каким образом вы реализовали защиту приложения.

- б. Лицензирование представляет собой процесс, осуществляемый после создания защиты программы. В то время как защита разрабатывается и реализуется однократно, лицензирование с помощью системы HASP HL отражает меняющийся со временем процесс. Таким образом, при разработке или изменении схемы лицензирования нет необходимости в изменении защиты приложения. Система обеспечит вас всеми необходимыми инструментами для изменения и обновления лицензий вашего программного обеспечения.

### **Функции**

В системе HASP HL под функцией понимается как все приложение, так и его исполняемый файл, или просто определенная функция («Print», «Save» или «Draw»). Функция представляет собой часть вашего программного обеспечения, доступ к которой вы хотите контролировать.

## **Лицензии**

Лицензии — это комбинация условий, закрепленных за функцией, а также сама функция внутри защищенного с помощью HASP HL приложения. Лицензии создаются с помощью программного обеспечения HASP HL и хранятся на аппаратном ключе HASP HL.

## **Обновления**

Обновления — это изменения, применяемые к лицензиям, хранящимся на ключе HASP HL. Обновления могут изменять или дополнять лицензионную информацию. Обновления могут программироваться удаленно — нет необходимости отзываться ключи.

## **Одновременное использование**

Одновременное использование — это особый термин, связанный с условиями лицензирования в сетевой среде. Одновременное использование подразумевает возможность доступа к лицензируемой функции в одно и то же время несколькими пользователями. Одновременное использование контролируется особой утилитой — HASP HL License Manager, которая устанавливается на месте работы конечного пользователя (см. раздел «HASP HL License Manager»).

## Доступные средства лицензирования HASP HL

В данном разделе рассматриваются программное обеспечение и аппаратные средства системы HASP HL, применяемые при лицензировании защищенного программного обеспечения.

### Аппаратные средства

В таблице 8.1. описываются модели ключей HASP HL, которые могут использоваться при лицензировании.

Таблица 8.1 Лицензионные возможности аппаратных средств HASP HL

Модель ключа	Максимальное кол-во лицензий	Возможности
HASP HL Pro	16	Активации Память
HASP HL Max	112	Активации Память
HASP HL Time	112	Активации Память Встроенные часы
HASP HL Net	112	Активации Память Сетевая среда

### Программное обеспечение

Частью системы HASP HL является современное программное обеспечение, предназначенное для создания и управления лицензиями. Для лицензирования используется следующее программное обеспечение:

- **HASP HL Factory** – является частью комплекта утилит Vendor Center. Данный инструмент следует использовать для задания и применения лицензий к защищенному программному обеспечению (см. также раздел «Лицензирование с помощью HASP HL Factory»).
- **HASP HL RUS** – эта утилита устанавливается на месте работы конечного пользователя. Она необходима для обновления лицензий HASP HL (см. также раздел «Система дистанционного перепрограммирования»).

- **HASP License Manager** — используется только с ключами HASP HL Net. Данное приложение передается конечному пользователю (см. также «Работа с HASP License Manager»).

## Планирование процесса лицензирования с помощью HASP HL

Процесс лицензирования с помощью HASP HL требует предварительного планирования.

HASP HL Factory позволяет определить элементы лицензирования и использовать эти элементы в разрабатываемых вами схемах лицензирования. Следует рассмотреть следующие вопросы:

1. Лицензирование каких функций необходимо предусмотреть?
2. Как будет осуществляться лицензирование этих функций?
3. Как эти функции могут быть скомбинированы в целях увеличения продаж?

## Выделение функций из программного обеспечения

Первым шагом при планировании лицензирования является выделение функциональных компонентов вашего программного обеспечения. Большинство программных продуктов можно разделить на несколько функциональных компонентов.

Такой шаг является важной частью в подготовке схемы лицензирования. Мы рекомендуем составить перечень всех функций, которые вы хотели бы лицензировать.

Например, компания, разрабатывающая программное обеспечение для строительных фирм, определяет три особые функции, подлежащие лицензированию:

- Печать отчетов — PRINT REPORT
- Сохранение отчетов - SAVE
- Генерирование проектных отчетов —REPORT GENERATOR

Данным функциям присваиваются уникальные номера (см. таблицу 8.2.).

Таблица 8.2 Пример перечня функций

Перечень функций	
Функция	Присвоенный номер
PRINT REPORT	34
SAVE	4
REPORT GENERATOR	48

Для работы с этим списком следует использовать HASP HL Factory (см. раздел «Присвоение функциям идентификаторов»).

### Сгруппируйте функции в «Пакеты»

После присвоения функциям идентификаторов, вы можете создать пакеты, являющиеся комбинацией этих функций. Для создания пакетов следует использовать HASP HL Factory (см. также раздел «Пакеты»).

Например, вы можете создать пакет «Оценочный», предназначенный для ознакомления с возможностями вашего программного обеспечения. В этом случае вы можете включить в пакет минимальное число функций. Вы также можете создать пакет «Корпоративный», включающий в себя все функции и предназначенный для больших корпоративных пользователей. Можно также создать промежуточный вариант — пакет «Обычный пользователь», в котором предлагается больше функций, чем в пакете «Оценочный», но меньше, чем в пакете «Корпоративный».

С помощью HASP HL Factory вы задаете содержание пакетов, но количество вариантов пакетов и составляющие их компоненты определяются вами.

Таблица 8.3 Примеры пакетов и их содержание

Функция	«Оценочный»	«Обычный пользователь»	«Корпоративный»
PRINT REPORT	Включена Условия: 5 активаций на лицензию	Включена Условия: 500 активаций на лицензию	Включена Условия: неограниченное использование
SAVE	Включена Условия: 5 активаций на лицензию	Включена Условия: неограниченное использование	Включена Условия: неограниченное использование
REPORT GENERATOR	Не включена	Включена Условия: достижение срока окончания лицензии	Включена Условия: до 250 станций, одновременно работающих с функцией

В таблице 8.3. приводятся возможные варианты формирования пакетов. Функция REPORT GENERATOR не доступна в оценочном пакете. Пакеты «Обычный пользователь» и «Корпоративный» содержат одинаковые функции, однако условия лицензирования различаются.

## Функции лицензирования

HASP HL позволяет контролировать функции с помощью License Forms (формы лицензий). Эти формы налагают на использование функции определенные условия. Вы можете контролировать использование функции с помощью:

- установки счетчика активаций функции;
- установки даты окончания лицензии на функцию;
- контроля за числом одновременного использования функции.

Каждый способ контроля функции имеет в HASP HL Factory определенную форму лицензии.

В таблице 8.3 показано, как воображаемая компания определяет условия использования функций, включенных в различные дистрибуционные пакеты. Хотя пакеты «Обычный пользователь» и «Корпоративный» включают в себя одинаковые функции, условия их применения различаются.

Ключ HASP HL Net, приложенный к «Корпоративному» пакету, обеспечит контроль над количеством обращений к функции REPORT GENERATOR, а ключ HASP HL Time — над сроком истечения лицензии на использование этой функции в пакете «Обычный пользователь».

Выбор модели ключей HASP HL зависит от того, каким именно образом вы хотите ограничить и контролировать использование лицензируемой функции.

## Лицензирование с помощью HASP HL в действии

В данном разделе описываются шаги и применение инструментов системы HASP HL, позволяющие реализовать выбранную вами схему лицензирования.

### **Шаг 1. Используйте HASP HL Factory, чтобы определить лицензируемые элементы**

После разработки схем лицензирования с помощью HASP HL Factory вы можете определить функции, условия лицензирования и пакеты. Данный инструмент также поможет вам в выборе наиболее подходящей модели аппаратного ключа HASP HL.



Определенный вами в HASP HL Factory перечень функций следует передать вашим разработчикам программного обеспечения. Данный перечень будет выполнять роль справочника при назначении идентификаторов каждой функции. Более подробная информация по идентификаторам функций приводится в разделе «Функция Login в HASP HL API»

### **Шаг 2. Используйте HASP HL Factory для определения команд**

С помощью HASP HL Factory, используя перечень лицензируемых элементов, создайте команды. Команды HASP HL Factory заносят условия лицензирования на ключи HASP HL (смотри также раздел «Определение команд»).

### **Шаг 3. Используйте HASP HL Factory для выполнения команд**

Данные о лицензировании, содержащиеся в команде, хранятся на ключе (см. также раздел «Выполнение команд»).

### **Шаг 4. Доставка программного обеспечения и ключей**

Доставьте программное обеспечение и несущие лицензии ключи вашим клиентам. Не забудьте обеспечить их необходимыми драйверами для

работы ключей HASP HL на их системах (см. также раздел «Поставка драйверов и демонов HASP HL»). В поставляемый клиентам пакет также необходимо включить программу HASP License Manager (см. раздел «HASP License Manager»).

### **Шаг 5. Обновите лицензии**

Система HASP HL позволяет обновлять лицензии непосредственно на рабочих местах пользователей. Более подробно эта тема рассмотрена в разделе «Система дистанционного перепрограммирования». Используйте утилиту HASP HL RUS для получения информации по уже переданным ключам HASP HL, а также для передачи обновлений, созданных с помощью HASP HL Factory (см. также раздел «Выполнение команд с помощью Дистанционной системы перепрограммирования»).



# Лицензирование с помощью HASP HL Factory

В главе описывается процесс создания, управления и реализации схем лицензирования с помощью HASP HL Factory. В главе рассматриваются следующие темы:

- условия применения HASP HL Factory;
- структура данных HASP HL Factory;
- создание команд HASP HL Factory;
- выполнение команд HASP HL Factory.

## Введение

HASP HL Factory является одной из программ, включенных в состав Vendor Center. Основная задача HASP HL Factory заключается в создании лицензий, которые в дальнейшем переносятся на ключи HASP HL. В задачи этой программы также входит непосредственно перенос данных в память ключа HASP HL и создание файлов обновлений, данные из которых переносятся затем на ключи пользователей.

В отличие от инструментов, используемых для создания защиты приложений (см. разделы «Защита с помощью HASP HL» и «Защита с помощью HASP HL API»), HASP HL Factory не взаимодействует с какими-либо компонентами вашего программного обеспечения.

Будучи исключительно инструментом, для работы с лицензиями, HASP HL Factory применяется только для определения лицензируемых элементов защищенных приложений, для переноса лицензий на ключи, а также для записи особых данных в память ключей.

HASP HL позволяет просматривать лицензии, хранящиеся на ключе HASP HL, либо напрямую (если ключ подключен к вашему компьютеру), либо удаленно (с использованием возможностей Системы дистанционного перепрограммирования).

В вашу версию HASP HL Factory включен пример, позволяющий разобраться в структуре данных различных элементов лицензирования. Программа также снабжена справочной системой.

## Условия применения HASP HL Factory

Существует три основных условия успешного применения HASP HL Factory:

- программное обеспечение;
- аппаратное обеспечение;
- предварительное планирование схем лицензирования.

## Требования к программному обеспечению

На вашей системе должны быть установлены следующие программы и компоненты:

- Драйверы или демоны HASP HL.
- Программное обеспечение HASP HL, включая Vendor Center.
- В папке *Vendorcodes* должен храниться требуемый Код разработчика (см. раздел «Получение кода разработчика с мастер-ключа HASP HL»). При использовании ключа HASP HL Demo, входящего в Демонстрационный Комплект, Код разработчика DEMOMA будет доступен всегда.
- Одна из следующих операционных систем: Windows 2000/XP или более поздняя версия Windows.

## Требования к аппаратному обеспечению

Если вы хотите применять HASP HL Factory для выполнения команд, вам потребуются:

- Мастер-ключ HASP HL с уникальной Серией ключа. Работа с новыми мастер-ключами HASP HL может быть начата с помощью HASP HL Factory.



Если вы оцениваете возможности HASP HL Factory, для выполнения команд используйте имеющийся пример (Sample). Пример имеет интегрированную серию и не требует мастер-ключа HASP HL.

- Ключ HASP HL, в который вы запишите команду. Этот ключ должен отвечать требованиям выбранной вами схемы лицензирова-

ния. Информация по лицензионным возможностям различных ключей HASP HL содержится в таблице 8.1.

## **Требования к планированию**

Как уже было сказано ранее, схему лицензирования необходимо продумать до применения HASP HL Factory. Более подробная информация по этому вопросу приводится в разделе «Планирование процесса лицензирования с помощью HASP HL».

После проведения оценки требований к схеме лицензирования, для изучения структур данных HASP HL Factory рекомендуется использовать пример.

## Структуры данных HASP HL Factory

Приведенные ниже элементы являются основными структурами данных HASP HL Factory:

- серии ключей;
- функции;
- пакеты;
- команды.

### Серии ключей

При заказе ключей HASP HL вашей компании присваивается Серийный номер, используемый для обработки ваших последующих заказов. Все ключи одной серии подвергаются процедуре уникального кодирования. Таким образом, ключи одной серии имеют уникальное поведение при шифровании и дешифровании.

Серии ключей используются HASP HL Factory для формирования функций и пакетов, а также для выполнения команд. Обычно для лицензирования всех ваших продуктов необходим один серийный номер. Тем не менее, запись о новой серии добавляется в HASP HL Factory с каждым подключением к компьютеру мастер-ключа HASP HL с новым Серийным номером. Эта запись включает:

- Серию ключа. Уникальный код из пяти-восьми символов, присвоенный определенному клиенту; используется при заказе ключей. Серия печатается на наклейках всех ключей HASP HL.
- Код разработчика. Уникальный код, требуемый системой HASP HL, для создания защиты или лицензирования (см. раздел «Получение Кодов Разработчика»).

Мастер-ключи HASP HL с новыми сериями ключей определяются при запуске HASP HL Factory. Вы также можете начать использовать новый мастер-ключ в процессе работы HASP HL Factory (для этого вам нужно подключить к компьютеру мастер-ключ HASP HL с новой серией).

После того, как серийный номер будет зарегистрирован HASP HL Factory, вы можете создавать функции и пакеты, связанные с этой серией.

### Функции

Наряду с пакетами, функции являются лицензируемыми элементами защищенных приложений. HASP HL Factory позволяет задавать и контролировать использование функций, являющихся частью вашего программного обеспечения.

Функции в HASP HL Factory отображаются как часть каждой серии.

---

## Присвоение функциям идентификаторов

При определении функции в HASP HL Factory вы присваиваете ей уникальный числовой идентификатор и имя. Однажды занесенная в систему функция отображается именем и номером. В HASP HL Factory любые две функции не могут иметь одинакового номера. Обычно, присвоенный функции номер не меняется в течение всей жизни продукта.

### Доступные идентификаторы

В таблице 9.1 приводится список доступных номеров каждой модели ключей HASP HL. Нулевое значение зарезервировано за функцией по умолчанию (прозрачная функция, доступная в случае подключения ключа HASP HL). Функцию по умолчанию не нужно конфигурировать с помощью HASP HL Factory, ее выполнение определяется подключенным ключом HASP HL.

**Таблица 9.1 Доступные программные номера различных моделей ключей HASP HL**

Модель ключа HASP HL	Программные номера
HASP HL Basic	-
HASP HL Pro	1-16
HASP HL Max	1-112
HASP HL Net	1-112
HASP HL Time	1-112

После того, как функции будут занесены в HASP HL Factory, вы можете формировать из них пакеты.

## Пакеты

Пакет может являться комбинацией следующих элементов:

- функции;
- условия лицензирования включенных функций;
- данные, записанные в память ключа HASP HL.

Пакеты можно сформировать в HASP HL Factory, что ускоряет процесс лицензирования защищенных приложений. Пакеты в HASP HL Factory отображаются как составные части серии ключа.

## Создание пакетов

Для создания пакета в HASP HL Factory необходимо задать его имя. Затем можно переходить к одному из следующих действий:

- Включить определенную функцию в пакет. Эта функция становится частью свойств пакета.
- Задать данные, которые будут записаны в память ключа HASP HL. Данные, содержащиеся в памяти ключа, становятся частью свойств пакета.

## Условия лицензирования функций

При добавлении в пакет для функции необходимо задать условия использования. Выбор условий лицензирования определяют ваши планы (см. раздел «Планирование процесса лицензирования с помощью HASP HL»).

### Формы лицензирования

Условия лицензирования реализуются в HASP HL Factory с помощью License Forms (Форм Лицензирования). В данной утилите доступны следующие формы лицензирования:

- **Counter.** Используется для контроля над числом активаций (запусков) функции.
- **Timer.** Используется для контроля над датой истечения срока действия лицензии.
- **Net.** Используется для контроля одновременного использования функции в сети.

Вне зависимости от типа выбранной формы, реальные условия лицензирования определяются тремя возможными способами:

- Используйте **Set (Установить)**, если вы хотите заменить существующие на ключе HASP HL лицензии на новые. **Set** следует применять к пакетам, в которых вы хотите, чтобы условия лицензирования отражали определенное значение.
- Используйте **Add (Добавить)**, если вы хотите добавить новые условия лицензирования к уже записанным на ключ HASP HL. **Add** следует применять к пакетам, в которых вы хотите, чтобы условия лицензирования добавлялись, а не записывались поверх.
- Используйте **Remove (Удалить)**, чтобы удалить определенное условие лицензирования функции с ключа HASP HL. **Remove** обычно применяется для отключения определенной лицензированной функции.



В системе лицензирования HASP HL вечная лицензия (**perpetual license**) отражается неограниченным счетчиком (**unlimited counter**).

### Данные, хранящиеся в памяти ключа

Для определения данных, которые будут записываться в память ключа HASP HL, следует использовать редактор, являющийся частью HASP HL Factory. Такие данные могут являться частью пакета. Вводимые в память данные могут записываться по различным адресам, что позволяет записывать несколько частей данных в различные области памяти. Таким образом, в памяти ключа HASP HL могут храниться разные данные для разных пакетов.

Вы можете вводить данные в двоичном коде или символами ASCII. Запись данных в память может использоваться для:

- присвоения уникального кода каждому пользователю;
- хранения паролей, программного кода, переменных и прочих данных.



В памяти ключа может содержаться несколько блоков данных, что позволяет использовать их несколькими пакетам.

### Команды

Команда представляет собой лицензионные данные, перенесенные на ключ HASP HL. Лицензионные данные содержат определение того, как контролируются защищаемые функции и реализуются изменения в памяти ключа. Когда команда применяется к ключу в первый раз, ключ инициализируется и активирует лицензионную схему, описываемую командой.

Команды в HASP HL Factory имеют три возможные формы:

- Команды, которые могут применяться к любому ключу HASP HL, отвечающему требованиям схемы лицензирования. Такие команды применяются к «пустому» ключу HASP HL. В этом случае, все данные, если они были до этого записаны на ключ, будут удалены.
- Команды, содержащие обновления для определенных ключей HASP HL. Если ключи находятся в вашем распоряжении, для прямого занесения обновлений вы можете использовать HASP HL Factory.

- Команды, содержащие обновления для определенных ключей HASP HL, которые находятся у пользователей. Такие команды приводятся в действие с помощью Системы дистанционного перепрограммирования.



HASP HL Factory можно применять для выполнения множественных команд. Для этого надо просто выбрать требуемые команды.

HASP HL задает очередность и показывает команды, которые надо выполнить, что отделяет их выполнение от создания.

После выполнения команды отслеживаются внутренней базой данных HASP HL Factory и больше не показываются.

---

## Создание команд HASP HL

### Условия

Для создания команды вам потребуется:

- Мастер-ключ HASP HL. Для оценки возможностей создания и выполнения команд вы можете использовать Sample.
- Ваша Серия ключа, заданная в HASP HL Factory. Для оценки вы можете использовать Sample.
- Ключ HASP HL, для которого вы будете выполнять команду.
- Дополнительно: функции и пакеты, заданные в HASP HL Factory.



В целях ускорения создания и выполнения команд в HASP HL Factory, мы рекомендуем задать лицензируемые элементы (пакеты и функции) заранее.

### Стадии

Процесс управления командами в HASP HL Factory является двухфазным.

1. Определение команды.
2. Выполнение команды.



Наличие двух фаз усиливает различия между созданием и обработкой команд.

### Создание команды

При создании в HASP HL Factory команде присваиваются:

#### Серия ключа

Обозначение серии, для которой создается команда. Все пакеты и функции, заданные для данной серии, могут использоваться при создании команды.

## Имя

Идентификатор, используемый для опознавания команды. Обычно эту роль выполняет имя ее получателя.



Ключи, содержащие информацию о лицензировании, распознаются системой, а их имена выводятся при подключении к компьютеру с запущенной утилитой HASP HL Factory.

## Комментарий

Это дополнительное свойство, содержащее описание команды.

## Типы команд

HASP HL Factory может использоваться для создания трех типов команд.

- a. Начальная команда. Данные команды содержат начальные лицензии, которые переносятся на ключ перед отправкой клиенту вместе с защищенным программным обеспечением. Этот тип команд инициализирует лицензирование и память ключа HASP HL.
- b. Обновление для подключенных ключей. После того, как ключ был инициализирован, содержимое его памяти и хранящиеся на нем лицензии могут быть изменены. Для этого ключ должен быть подключен к компьютеру с запущенной утилитой HASP HL Factory. Вы можете добавить, обнулить или удалить текущий статус лицензии на ключе.
- c. Обновления, применяемые к ключам HASP HL, находящимся в распоряжении пользователей. HASP HL Factory создает обновление лицензии, а конечный пользователь должен самостоятельно его применить.

## Добавление функции

Если функции уже заданы в HASP HL Factory, вам остается только добавить их в команды.

В зависимости от ваших планов лицензирования, при добавлении функции к команде, необходимо указать условия лицензирования функции (см. раздел «Планирование процесса лицензирования с помощью HASP HL»).

---

## Формы лицензирования

Условия лицензирования применяются в HASP HL Factory с помощью Форм Лицензирования. В данной утилите доступны следующие формы лицензирования:

- **Counter.** Используется для контроля над числом активаций функции.
- **Timer.** Используется для контроля над датой истечения срока действия лицензии.
- **Net.** Используется для контроля одновременного использования функции в сети.

Вне зависимости от типа выбранной формы, реальные условия лицензирования определяются тремя возможными способами:

- Используйте **Set (Установить)**, если вы хотите заменить существующие на ключе HASP HL лицензии на новые. **Set** следует применять к пакетам, в которых вы хотите, чтобы условия лицензирования отражали определенное значение.
- Используйте **Add (Добавить)**, если вы хотите добавить новые условия лицензирования к уже записанным на ключ HASP HL. **Add** следует применять к пакетам, в которых вы хотите, чтобы условия лицензирования добавлялись, а не записывались поверх.
- Используйте **Remove (Удалить)**, чтобы удалить определенное условие лицензирования функции с ключа HASP HL. **Remove** обычно применяется для отключения определенной лицензированной функции.

## Добавление пакета

К серии ключа, для которого создана команда, можно только добавить пакеты, определенные в HASP HL Factory. После включения пакета в команду, все его содержимое (функции, условия лицензирования, записи в память электронного ключа) становится частью команды. Все данные, включенные в команду, можно редактировать.

## Запись данных в память ключа и команду

Для определения данных, которые будут записаны в память ключа HASP HL, следует использовать редактор HASP HL Factory. Данные можно определить как часть команды. Ввод данных может осуществляться в двоичной системе и символами ASCII. Запись данных в память может использоваться для:

- присвоения уникального кода каждому пользователю программного обеспечения;
- сохранения паролей, программного кода, переменных и иных данных.



Редактор памяти отображает все содержимое памяти ключа HASP HL. В данном случае, редактор отражает раздел памяти, который был изменен при занесении на ключ сведений о пакете.

## Выполнение команд

Существует три способа использования HASP HL Factory при работе с командами:

- Вы можете выполнить команду, которая в первый раз инициализирует ключ HASP HL. Обычно такой ключ еще не передавался пользователю. Эта команда не является обновлением существующей лицензии.
- Вы можете применить команду непосредственно к ключу HASP HL, если он подключен к компьютеру с запущенной утилитой HASP HL Factory. Эта команда является обновлением существующей лицензии.
- Вы можете создать обновление, которое применяется с помощью утилиты HASP HL RUS. Эта команда является обновлением существующей лицензии.

## Применение команды к подключенному ключу HASP HL

После помещения вашей команды в список задач, ее можно применить к подключенному ключу HASP HL. Команду до ее выполнения можно только изменить. После выполнения команды, она перестает отображаться и помещается во внутреннюю базу данных HASP HL Factory.

Убедитесь в том, что ключ, к которому применяется команда, отвечает схеме лицензирования. Например, если команда содержит лицензию на функцию, которая имеет ограничение по сроку действия, следует использовать ключ HASP HL Time.

### Процедура

После создания команды ее можно выполнять. Для этого вам потребуется:

- мастер-ключ HASP HL для серии, под которую создавалась команда;
- ключ HASP HL — для которого выполняется команда.

Для выполнения команды:

1. Подключите мастер-ключ HASP HL для серии, под которую создавалась команда.

2. Подключите ключ HASP HL, для которого будет выполняться команда.



Для выполнения команды, содержащей обновление для определенного ключа, вам следует подключить именно этот ключ. Для выполнения команды на «пустом» ключе вы можете подключить любой ключ HASP HL, соответствующий требованиям избранной схемы лицензирования, определяемой командой.

3. Из списка, показываемого HASP HL Factory, выберите команду, которую вы хотите выполнить.
4. Выполните команду.

## Выполнение команды с помощью Системы дистанционного перепрограммирования

Если вам необходимо обновить лицензию, хранящуюся на ключе пользователя, вам следует использовать HASP HL Factory для создания обновления лицензии. HASP HL Factory позволяет создавать обновления лицензий в двух форматах, используемых Системой дистанционного перепрограммирования:

- Исполняемый файл *hasprus.exe*, запускаемый пользователем.
- V2C-файл (см. раздел «Утилита HASP HL RUS»).

Для получения более подробной информации по процессу работы системы HASP HL с обновлениями для ключей удаленных пользователей, обратитесь к разделу «Система дистанционного перепрограммирования».

## Просмотр ключей HASP HL

HASP HL Factory позволяет просматривать статус ключа HASP HL и содержимое его памяти после того, как к нему была применена команда. Для этого ключ должен быть подключен к компьютеру.

Чтобы просмотреть содержимое ключа HASP HL:

1. Подключите ключ.
2. В меню **Tools** выберите **View Connected Key**.
3. Окно **Key Properties** содержит следующую информацию:
  - **Batch (Серия)** — идентификатор серии, использованной для создания лицензии, перенесенной на ключ.
  - **Name (Имя)** — идентификатор последней команды, примененной к ключу.

- **HASP ID** — уникальный серийный номер ключа.
- **Features (Функции)** — выберите вкладку **Features**. Будут отображены включенные функции.
- **Memory (Память)** — для просмотра памяти ключа, выберите вкладку **Memory**.

Вы также можете проверить содержимое переданного пользователю ключа с помощью C2V-файла, содержащего информацию о статусе лицензии (см. раздел «Применение HASP HL RUS»).

## Создание обновления для определенного ключа

HASP HL Factory можно применять для создания обновления лицензии непосредственно для подключенного ключа.

1. откройте просмотр свойств ключа;
2. щелкните по **Create Update**.

Создание обновления инициирует создание команды для определенного ключа. Команда автоматически загружает текущее содержимое ключа. Вы можете выбрать редактирование данных или добавление новых данных на ключ.



Если вы удалите с ключа функцию, обновление не коснется ее. Все функции, измененные в редакторе команды, записываются при выполнении команды обновления поверх существующих на ключе функций. Для удаления лицензии к функции следует применять **Remove**.

После подготовки команды для применения ее к ключу можно использовать HASP HL Factory. Также для работы с ключом удаленного пользователя можно сгенерировать V2C-файл, содержащий информацию об обновлении лицензии.

## Считывание времени с ключей HASP HL

До передачи ключей HASP HL Time, содержащих информацию о лицензиях, пользователям, с помощью HASP HL Factory с них можно считать или установить на них время. Для получения более подробной информации о том, как это можно сделать, обратитесь к справочной системе HASP HL.



Часы HASP HL предустановлены на универсальное время UTC.

# Система дистанционного перепрограммирования (RUS – Remote Update System)

В главе рассматриваются следующие темы, связанные с описанием работы Системы дистанционного перепрограммирования HASP HL RUS:

- компоненты HASP HL RUS;
- принципы работы HASP HL RUS;
- применение утилиты *hasprus.exe*;
- конфигурация V2C-файла.

## Концепция

HASP HL RUS является мощной системой, позволяющей осуществлять безопасное удаленное обновление ключей HASP HL, находящихся у конечных пользователей. С помощью данной программы вы можете обновлять содержимое памяти следующих моделей ключей HASP HL: HASP HL Pro, HASP HL Max, HASP HL Net и HASP HL Time.

HASP HL RUS представляет собой простое и безопасное средство управления лицензиями, находящимися в памяти ключей HASP HL, даже после того, как вы передали их вместе с защищенным программным обеспечением пользователям. Вам просто следует создать обновление лицензии и послать файл обновления конечному пользователю, например, по почте. HASP HL RUS также позволяет вам получить информацию по статусу лицензий с мест работы пользователей. Система позволяет даже манипулировать лицензиями на функции на месте работы клиента.

В качестве части основной концепции, лежащей в основе работы системы HASP HL, программа позволяет осуществлять дальнейшее

лицензирование уже после реализации защиты (см. раздел «Автоматическая система лицензирования»).

Применение HASP HL RUS позволяет не отзываться ключи у клиентов в том случае, если возникает необходимость внесения изменений в условия лицензирования функций.

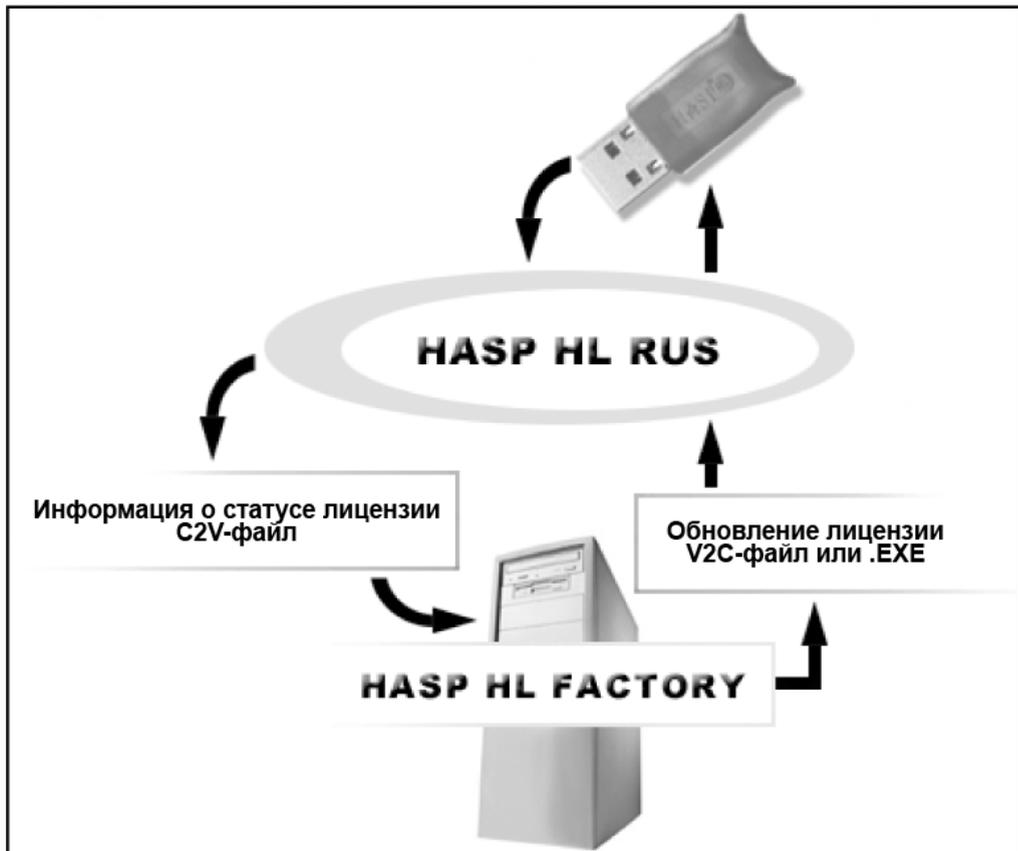
## Компоненты

HASP HL RUS состоит из следующих компонентов:

- **HASP HL Factory.** Данный инструмент разработчика применяется для создания обновлений, которые в дальнейшем доставляются пользователю. Программа также необходима для открытия посылаемых пользователями файлов, которые содержат информацию о текущем статусе ключей. Информация об обновлении создается тем же инструментом, что и первоначальная лицензия.
- **HASP HL RUS.** Программа представляет собой утилиту, осуществляющую обновление лицензий, которая передается конечным пользователям. Утилита — *hasprus.exe* — используется для сбора информации о статусе лицензии на ключе конечного пользователя, а также для обновления лицензий. Эта утилита настраиваемая; ее генерирование осуществляется HASP HL Factory.
- **C2V-файлы.** Файлы, передаваемые клиентом разработчику. Эти файлы имеют расширение *.c2v* и содержат информацию о состоянии лицензий на переданных клиентам ключах. Для создания C2V-файла и сохранения информации клиентам необходимо нажать всего одну кнопку после запуска *hasprus.exe*. Содержимое такого файла можно просмотреть с помощью HASP HL Factory. Кроме того, C2V-файлы содержат образ содержимого памяти ключа HASP HL, находящегося у клиента.
- **V2C-файлы.** Файлы, передаваемые разработчиком клиенту. Эти файлы содержат информацию об обновлениях к лицензиям. Данные файлы создаются с помощью HASP HL Factory и имеют две формы: исполняемые файлы *.exe* или файлы с расширением *.v2c*, которые открываются с помощью HASP HL RUS. В дополнение к лицензионной информации в эти файлы вы также можете добавить данные, которые будут перенесены в память ключа HASP HL пользователя.

Взаимосвязь компонентов, составляющих систему HASP HL RUS, представлена на рисунке 10.1.

Рисунок 10.1 Компоненты HASP HL RUS

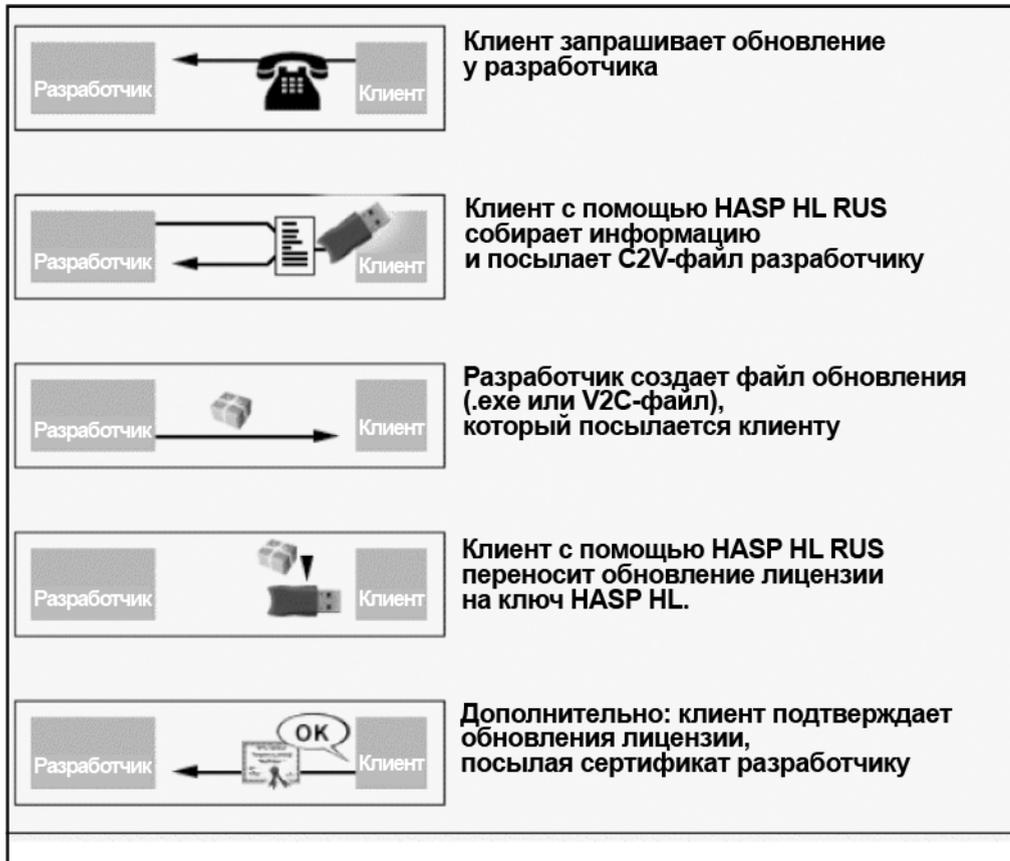


## Порядок работы с HASP HL RUS

Описанные в предыдущем разделе инструменты обеспечивают безопасность и простоту работы. На рисунке 10.2. показан пример применения HASP HL RUS.

Клиент хочет обновить лицензию функции и связывается с разработчиком программного обеспечения. Для создания обновления разработчику необходим C2V-файл, описывающий содержащуюся на ключе информацию. В данном файле содержится текущий статус лицензии, а также содержимое памяти ключа HASP HL.

Рисунок 10.2



Собранные данные позволяют разработчику создать обновление, удовлетворяющее требованиям клиента. Потребность в перенастройке защиты на месте работы конечного пользователя отсутствует. Разработчик создает обновление (V2C-файл или исполняемый файл), который далее

пересылается конечному пользователю. Конечный пользователь применяет обновление.

Разработчик может также задать в V2C-файле необходимость в послылке конечным пользователем подтверждения об осуществлении обновления. Такое подтверждение создается на месте работы пользователя после применения обновления и посылается конечным пользователем разработчику.

## Утилита HASP HL RUS

HASP HL RUS является программой, передаваемой конечному пользователю. В ее задачи входит:

- Сбор данных о текущем состоянии лицензии (включая содержимое памяти) с электронных ключей, переданных пользователям.
- Применение обновлений, посланных клиенту разработчиком.

Для создания и настройки утилиты следует использовать HASP HL Factory. Вы можете включить копию утилиты в состав программного обеспечения, передаваемого конечным пользователям.

HASP HL RUS можно передавать пользователям и отдельно — уже после создания обновления лицензии на функцию защищенного программного обеспечения. HASP HL Factory может преобразовать V2C-файл в исполняемый файл. Для получения подробной информации о различных возможностях создания утилиты HASP HL RUS с помощью HASP HL Factory вам следует обратиться к справочной системе HASP HL Factory.

Утилита HASP HL RUS имеет графический интерфейс, разделенный на две вкладки:

- **Collect Key Status Information.** Вкладка используется для сбора информации по текущему статусу лицензий, находящихся на ключе HASP HL. Пользователю необходимо указать имя и место для хранения генерируемого C2V-файла.
- **Apply License Update.** Данная вкладка используется для открытия V2C-файла и обновления лицензий, находящихся на ключе HASP HL.

При генерировании V2C-файла с помощью HASP HL Factory вы можете включить в интерфейс утилиты одну или обе вкладки.

## Применение HASP HL RUS

Утилита HASP HL RUS сделана таким образом, чтобы быть максимально понятной конечным пользователям. Осуществление сбора данных или обновления лицензии являются простыми в осуществлении процессами.

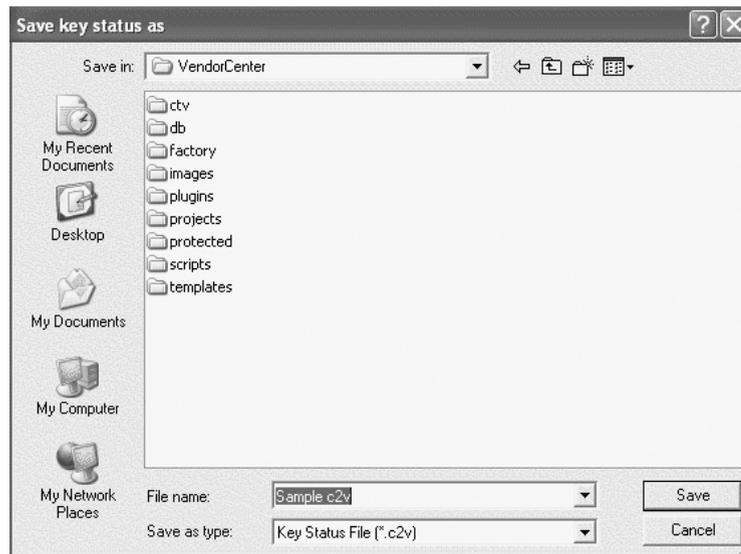
## Сбор данных о лицензии с ключа HASP HL

Чтобы включить всю необходимую информацию для поиска ключа HASP HL утилита HASP HL RUS может быть настроена с помощью HASP HL Factory.

Для сбора информации о текущем статусе лицензии, находящейся на ключе HASP HL, следуйте следующей процедуре:

1. Запустите *hasprus.exe*.
2. Выберите вкладку **Collect Key Status Information**.
3. Щелкните мышкой по кнопке **Collect information**. Появится окно **Save key status as** (рис. 10.3)

Рисунок 10.3 Экран «Save key status as...»



4. Укажите папку, в которой вы хотите сохранить C2V-файл. Введите имя файла и щелкните по кнопке **Save**.
5. Вы создали C2V-файл для ключа HASP HL. Теперь файл может быть послан для обработки и осуществления обновления.

## Применение обновления

Для применения обновления к лицензиям, хранящимся на ключе HASP HL, следует придерживаться следующей схемы:

1. Запустите *hasprus.exe* или щелкните мышью по V2C-файлу, содержащему данные об обновлении.



Если вы получили обновление в виде исполняемого файла, просто щелкните по нему мышью. Это приведет к автоматическому запуску HASP HL RUS.

2. Выберите вкладку **Apply License Update**. Если обновление имеет форму исполняемого файла, будет показана только эта вкладка.
3. Если поле **Update file** пустое, откройте папку, в которой находится файл обновления (V2C-файл).
4. Щелкните по кнопке **Apply update**. Новые данные о лицензии будут перенесены на ключ HASP HL.
5. Может появиться подсказка о необходимости создания сообщения, подтверждающего процедуру обновления лицензии. Пошлите это подтверждение для дальнейшей обработки.

## Маркировка V2C–файла

При применении утилиты HASP HL Factory для создания файла обновления, который в дальнейшем будет использоваться конечным пользователем, вы можете промаркировать V2C-файл.

В утилите HASP HL Factory есть HTML-редактор, позволяющий вставлять в данные инструкции или сообщения для конечного пользователя. Эта функция может использоваться для идентификации источника и содержания обновления.

С помощью HASP HL Factory можно промаркировать:

- V2C-файл. Этот файл будет загружаться пользователями с помощью HASP HL RUS.



Мы рекомендуем поставлять защищенное программное обеспечение вместе с маркированной версией HASP HL RUS.

- Исполняемый файл. Это исполняемый файл, связанный с утилитой HASP HL RUS. Конечный пользователь должен запустить исполняемый файл и следовать появляющимся на экране инструкциям.

Получить более подробную информацию о внесении изменений в V2C-файл можно обратившись к справочной системе HASP HL Factory.



# Передача HASP HL вместе с вашим программным обеспечением

В данной главе рассматриваются следующие темы, связанные со способами передачи необходимого программного обеспечения конечным пользователям:

- Какие компоненты программного обеспечения HASP HL должны передаваться пользователям?
- Возможные варианты передачи пользователям драйверов HASP HL для Windows.
- Возможности и способы передачи демона HASP HL для пользователей, работающих на платформах Mac и Linux.

## Программное обеспечение HASP HL, предназначенное для конечных пользователей

Установка системы HASP HL требует передачи определенного программного обеспечения пользователям. Оно должно устанавливаться на рабочих местах пользователей и обеспечивать корректность работы защищенного программного обеспечения.

Вам не нужно передавать пользователям все программное обеспечение, описываемое в данной главе.



Драйверы и демоны HASP HL должны обязательно входить в комплект передаваемого пользователям программного обеспечения.

## Программное обеспечение, связанное с защитой

Самой важной частью программного обеспечения, устанавливаемого на компьютере пользователя, являются драйверы и демоны HASP HL. Эти приложения необходимы для того, чтобы ключ мог обмениваться данными с защищенным приложением (см. раздел «Передача клиентам драйверов и демонов HASP HL»).

## Программное обеспечение для работы в сети

Если вы передаете конечным пользователям ключи HASP HL Net, вам также следует передать им следующие приложения:

- **HASP License Manager.** Программа, предназначенная для контроля использования защищенного и лицензированного приложения в сети (см. раздел «HASP License Manager»).
- **Aladdin Monitor.** Программа применяется в связке с HASP License Manager и ключами HASP HL Net (см. раздел «Aladdin Monitor»).

## Программное обеспечение для устранения неисправностей

Следует рассмотреть возможность передачи всем конечным пользователям программы Aladdin DiagnostiX. Эта утилита предназначена для сбора информации о ключах HASP HL и среды их работы (см. раздел «Диагностика ключей HASP HL»).

## Программное обеспечение, предназначенное для обновления лицензий

Утилита HASP HL RUS передается пользователям при осуществлении удаленного обновления лицензий, находящихся на переданных конечным пользователям ключах HASP HL (см. раздел «Утилита HASP HL RUS»).

## Передача клиентам драйверов и демонов HASP HL

На компьютерах конечных пользователей должны устанавливаться драйверы или демоны ключей HASP HL. Без них защищенное приложение не сможет обмениваться данными с ключом HASP HL.

В следующем далее разделе рассматриваются различные возможности передачи драйверов и демонов HASP HL конечным пользователям.

## Передача драйверов HASP HL под Windows

Варианты передачи драйверов HASP HL пользователям представлены ниже.

- Для загрузки драйверов из Интернета можно использовать Windows Update или Aladdin DiagnostiX. Для этого требуется соединение с Интернетом.

- Можно интегрировать установку драйверов HASP HL в процедуру установки защищенного приложения с помощью:
  - модулей слияния
  - HASP HL API для установки драйверов.
- Можно передать конечным пользователям одну из утилит HASP HL, устанавливающих драйверы:
  - *HASPUserSetup.exe* — программа установки с графическим интерфейсом
  - *haspdinst.exe* — программа имеет интерфейс командной строки

## Windows Update

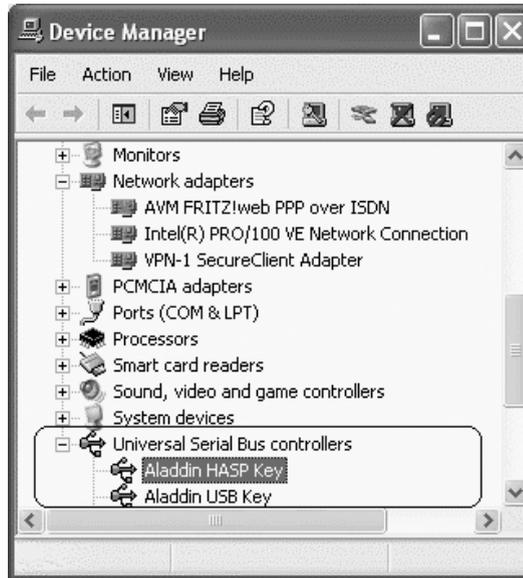
Если конечный пользователь установил защищенное приложение на систему с операционной системой Windows XP или 2003 и имеет доступ к Интернету, ему просто требуется подключить ключи HASP HL к своему компьютеру. Все драйверы HASP HL сертифицированы Microsoft и, таким образом, автоматически загружаются с сайта Microsoft Update.

После того, как ваши конечные пользователи подключат ключи HASP HL к компьютеру:

1. Система проинформирует их о том, что было обнаружено новое устройство.
2. Автоматически установится драйвер ключа HASP HL.
3. Когда на ключе HASP HL загорится индикатор, процесс установки можно считать завершенным.

Убедиться в том, что требуемые драйверы были установлены, можно с помощью Диспетчера Устройств Windows. Драйверы устройств Aladdin для каждого подключенного ключа HASP HL должны быть указаны в группе **Universal Serial Bus Controller**.

**Рисунок 11.1** Окно Диспетчера Устройств с драйверами устройств Aladdin



## Применение Aladdin DiagnostiX

После установки защищенного программного обеспечения и ключей HASP HL на своем рабочем месте пользователи могут воспользоваться функцией **Update Driver** программы Aladdin DiagnostiX. Для этого необходимо подключение к Интернету и установка *askdiag32.exe* на компьютере, где было установлено программное обеспечение.



В отличие от описанной ранее возможности обновления с помощью Windows Update, использование Aladdin DiagnostiX позволяет осуществлять обновление драйверов на платформах с более ранними версиями операционной системы, чем Microsoft Windows XP или 2003.

## Модули слияния

В целях незаметной интеграции установки драйвера HASP HL и установки защищенного программного обеспечения с помощью MSI можно использовать модули слияния HASP HL. Модули слияния включают совместно используемые компоненты Windows Installer® и логику установки в приложение. Совместно используемый код, файлы, ресурсы, записи в регистре и порядок установки объединены одним общим фай-

лом. Таким образом, установка драйвера HASP HL возможна с применением модуля слияния *haspds.msm*.



Модуль слияния *haspds.msm* не может быть запущен как отдельное приложение.

Модуль слияния *haspds.msm* находится в следующей папке установочного компакт-диска HASP HL:

Windows/Installed/API/DriverInstall/MSI

Пример модуля слияния можно найти в следующей директории:

Windows/Installed/Samples/DriverInstall/MSI

## Концепция

Интегрированный в ваш MSI-инсталлятор модуль *haspds.msm* копирует библиотеку *haspds\_windows.dll* в системную папку Win32 компьютера конечного пользователя. Далее эта библиотека вызывается модулем MSI при установке или удалении драйверов HASP HL.



Преимущества использования модулей слияния HASP HL и инсталлятора MSI:

1. Конечные пользователи получают один инсталляционный файл, который устанавливает и защищенное приложение, и драйверы HASP HL.
2. Повторное применение инсталлятора MSI позволяет устранить ошибки, возникшие при установке.

## Контрольные вопросы для реализации установки с помощью модулей слияния

До включения модулей слияния HASP HL в вашу программу установки необходимо убедиться в следующем:

- Для осуществления установки драйвера конечным пользователям будут необходимы права администратора. Это следует предусмотреть при создании установочного файла.
- Применение модулей слияния HASP требует Windows Installer версии 2.0 и выше.



Не применяйте более ранние версии Microsoft Installer!

- При установке драйверов HASP HL параллельно не должны запускаться требующие их присутствия процессы.
- Для ознакомления с действием модулей слияния HASP HL вы можете обратиться к примеру *haspds.msm* для MSI.

## Реализация

Применение модулей слияния HASP HL представляет собой простой процесс, требующий включения файла \*.msm, содержащего процедуру установки драйвера, в MSI-совместимый файл установки. После создания MSI-инсталлятора программа установки будет содержать модули слияния HASP HL.

Перед тем как применять модули слияния, следует обратиться к приведенному выше перечню контрольных вопросов.

## Пример модулей слияния

Перед тем как включать модули слияния в ваш файл установки, изучите пример инсталляционного файла с включенным в него модулем слияния HASP HL. Файл представляет собой MSI-совместимый инсталлятор с включенным в него модулем слияния и всеми необходимыми библиотеками для установки драйверов HASP HL.

Файл примера выполняет следующие действия:

- Проверяет у пользователя наличие прав администратора для установки драйверов HASP HL.
- Прекращает работу HASP License Manager до начала установки драйвера и запускает его по завершению установки.
- Устанавливает и удаляет драйверы HASP HL.

## API для установки драйвера HASP HL

Для интеграции процесса инсталляции в настраиваемый файл установки следует применять API для установки драйвера HASP HL. Подробная информация по этому вопросу содержится в файле помощи, который находится в папке со следующим путем:

Program Files/ Aladdin/HASP HL/Drivers/

Пример API для установки драйвера HASP HL можно найти на установочном диске HASP HL по следующему пути:

```
Windows/Installed/Samples/DriverInstall/C/
```

В целях обучения применению API для установки драйвера HASP HL можно использовать этот пример и файл помощи.

## ***haspdinst.exe***

Утилита *haspdinst.exe* является приложением командной строки, устанавливающим драйверы устройств HASP HL в операционных системах Windows 98/ME/2000/XP/Server 2003. Это приложение можно передавать конечным пользователям. Копия *haspdinst.exe* находится по следующему пути на установочном компакт-диске HASP HL:

```
Windows/Installed/Redistribution/Drivers/
```



Этот метод можно применять только в том случае, если вы уверены, что ваши конечные пользователи могут пользоваться интерфейсом командной строки.

### **Для установки драйверов устройств HASP HL:**

Напечатайте в командной строке `haspdinst -i`

Подробная информация по данной утилите содержится в разделе «Утилита *haspdinst.exe*».

Проверить установку драйверов устройств HASP HL можно запустив Диспетчер устройств Windows. В разделе **Universal Serial Bus Controllers** должны появиться соответствующие записи (см. рисунок 11.1).

## ***HASPUserSetup.exe***

Копия файла *HASPUserSetup.exe* находится в следующей директории установочного компакт-диска HASP HL:

```
Windows/Installed/Redistribute/Drivers
```

Это простая в применении программа, устанавливающая драйверы HASP HL в операционные системы Windows 98/ME/2000/XP/Server 2003. Процесс инсталляции проходит с помощью мастера установки. При запуске файла пользователем появляется окно приветствия (рисунок 11.2). Для установки драйверов пользователю остается только следовать появляющимся на экране инструкциям.

**Рисунок 11.2** Окно приветствия HASP UserSetup.exe.



Проверить установку драйверов устройств HASP HL можно запустив Диспетчер устройств Windows. В разделе **Universal Serial Bus Controllers** должны появиться соответствующие записи.

## Передача пользователям демонов HASP HL для MAC

Пользователям, работающим на платформах Mac OS X, следует вместе с защищенным программным обеспечением передавать демон HASP HL. Без установки демона системы конечных пользователей не смогут распознавать ключи HASP HL, а защищаемые приложения не смогут функционировать без доступа к аппаратным ключам.

Все программное обеспечение, предназначенное для конечных пользователей, работающих на системах под управлением Mac OS, находится на установочном компакт-диске HASP HL в папке MacOS/Redistribute. Программное обеспечение для Mac-систем также можно загрузить по следующей ссылке:

<http://www.hasp.com/downloads>

## Возможные варианты передачи пользователям демона Mac

Все программное обеспечение, требуемое для передачи демона, находится на установочном компакт-диске HASP HL по следующему пути:

MacOS/Installed/Redistribute/Runtime/

В данном разделе рассматривается несколько возможных способов передачи демона Mac вашим клиентам. Каждый метод имеет свои преимущества. Методы представлены по убывающей степени предпочтительности.

### Инсталляционный образ

Инсталляционный образ HASP *Installation.dmg* находится в поддиректории *dmg* установочного компакт-диска HASP HL. Образ не может изменяться, однако его преимущество заключается в том, что его можно сжать в легко загружаемый формат. Еще одним преимуществом является корректная установка привилегий пользователя.



Поместите *HASP Installation.dmg* на ваш FTP-сайт и сделайте его доступным для загрузки.

Процедура установки демона с помощью *HASP Installation.dmg* описана в разделе «Установка демона».

### Инсталляционный пакет

Инсталляционный пакет демона HASP HL — *AKSUSB Install.pkg.sit* — находится в поддиректории *pkg* установочного компакт-диска HASP HL.

1. Скопируйте содержимое папки *pkg* на локальный диск.

2. Для разархивирования *AKSUSB Install.pkg.sit* следует применить **Stuffit Expander**.
3. Запустите файл *AKS Install.pkg.sit*.  
Появится окно, показанное на рисунке 11.3. Следуйте появляющимся на экране инструкциям.

**Рисунок 11.3** Установка демона HASP HL



Инсталляционный пакет может также быть включен в любую мульти-пакетную установку. Для получения более подробной информации по этому вопросу вам следует обратиться к соответствующему документу Apple Developer по адресу:

<http://developer.apple.com>.

Для получения дополнительной информации по использованию скриптов обратитесь к файлу *readme.txt*, находящемуся в поддиректории *pkg*.

### **Исходные файлы инсталляционного пакета**

Исходные файлы (описываются в таблице 11.1), необходимые для создания инсталляционного пакета для установки демона HASP HL, находятся в поддиректории *pkg-source*.

**Таблица 11.1 Исходные файлы инсталляционного пакета для установки демона в среде Mac OS X**

Файл	Описание
<i>Aksusbd_install.pmsp</i>	Файл проекта для инсталлятора демона. Используется утилитой PackageMaker при создании пакетов.
<i>dinst_Resources.sit</i>	Сжатый файл, содержащий демон <i>aksusbd</i> , а также все загрузочные параметры. Содержимое файла может изменяться в соответствии с потребностями клиентов.
<i>dinst_Resources.sit</i>	Сжатый файл, содержащий фоновый образ инсталлятора и скрипты оболочки, применяемые для управления пред- и после-установочными процессами. Содержимое файла может изменяться в соответствии с потребностями клиентов.
<i>aksusbd)forceinstall.pmsp</i>	Файл проекта, используемый для принудительной установки. Все новые данные, связанные с демоном HASP HL, записываются поверх старых данных.
<i>dinsforce_Resources.sit</i>	Сжатый файл, используемый для обычной или принудительной установки демона HASP HL (новые данные записываются поверх старых). Файл не содержит скриптов оболочки, позволяющих управлять пред- и после-установочными процессами.

Для включения исходных файлов в ваш инсталлятор следует осуществить процедуру, описанную ниже:

1. Скопируйте содержимое папки *pkg-sources* на локальный диск.
2. Для разархивирования всех сжатых файлов, описанных в таблице 11.1, следует применить **Stuffit Expander**.
3. Включите файлы в инсталлятор.

Для получения более подробной информации по этому вопросу вам следует обратиться к соответствующему документу Apple Developer по адресу:

<http://developer.apple.com>.

Для получения дополнительной информации по исходным файлам обратитесь к файлу *readme.txt*, находящемуся в поддиректории *pkg-sources*.

Соответствующий сжатый файл может быть интегрирован с инсталлятором вашего защищенного программного обеспечения. Конечный пользователь автоматически установит демоны вместе с защищенным приложением. Вы также можете предусмотреть вывод окна, аналогичного показанному на рисунке 11.3.



Все файлы *aksusbd* автоматически переносятся в корневой том компьютера клиента. Если вы хотите это предотвратить, для установки демона вам следует применять инсталляционные пакеты.

### Инсталляционные скрипты

Инсталляционные скрипты находятся в поддиректории *script* установочного компакт-диска HASP HL. Эти скрипты следует скопировать и выслать пользователям, которые будут использовать их для установки демона из командной строки. Данные скрипты не настраиваются.



Этот метод установки можно применять только в том случае, если ваши конечные пользователи умеют пользоваться интерфейсом командной строки.

Дополнительную информацию по использованию скриптов можно получить обратившись к файлу *readme.txt*, находящемуся в поддиректории *Script*.

## Передача конечным пользователям демона HASP HL для Linux

Конечным пользователям, которые будут работать с защищенным приложением в среде Linux, следует передать демон HASP HL — *aksusbd*. Без установки демона системы конечных пользователей не смогут распознавать ключи HASP HL, а защищаемые приложения не смогут функционировать без доступа к аппаратным ключам.

Все программное обеспечение, предназначенное для конечных пользователей, работающих на системах под управлением Linux, находится на установочном компакт-диске HASP HL в папке *Linux/Installed/Redistribute*.

Программное обеспечение для Linux-систем также можно загрузить по следующей ссылке:

<http://www.hasp.com/downloads>

### **Формат передачи демона HASP HL**

В папке *Linux/Redistribute* находится демон HASP HL, предназначенный для передачи пользователям в следующих форматах:

- В виде RPM-пакета для платформ под управлением Red Hat и SuSE
- В виде исполняемого файла с инсталляционным скриптом

### **Интеграция демона HASP HL в инсталлятор вашего приложения**

Вы можете включить демон HASP HL в программу установки вашего программного обеспечения. Вам следует убедиться, что программа установки осуществляет следующие шаги:

1. Необходимые RPM-пакеты HASP HL или исполняемые файлы копируются во временную папку.
2. Программа установки запускает скрипт или команду на установку.

Более подробное описание процесса установки демона HASP HL приводится в разделе «Установка HASP HL в операционной системе Linux».



# HASP License Manager

В главе рассматриваются следующие темы, касающиеся HASP License Manager:

- HASP License Manager для Windows/Mac/Linux;
- настройка HASP License Manager;
- настройка приложений, защищаемых ключом HASP HL Net;
- настройка среды HASP HL Net.

## Обзор возможностей HASP License Manager

HASP License Manager ведет таблицу открытых сессий, в которой регистрируются все защищенные приложения, обращавшиеся к ключу HASP HL Net. В таблице указывается каждое защищенное приложение и станция, с которой оно было запущено. Приложение и станция, на которой приложение было запущено, находятся в таблице до тех пор, пока приложение не закроет сессию подключения к ключу HASP HL Net.

Данная таблица используется HASP License Manager для отслеживания числа одновременно открытых копий защищаемого приложения в сетевой среде. Основная задача таблицы заключается в том, чтобы максимальное число запусков защищенных функций или приложений не превысило разрешенного лицензией значения. По умолчанию таблица отслеживает до 250 приложений.

## HASP License Manager для Windows

HASP License Manager для Windows доступен в двух формах:

- Приложение для операционных систем Windows 98SE/ME/NT/2000/XP/Server 2003
- Служба для операционных систем Windows 2000/XP/Server 2003

HASP License Manager для Windows может осуществлять обмен данными с помощью протоколов TCP/IP и IPX. Протоколы могут загружаться и выгружаться с помощью графического интерфейса программы или с помощью настройки файла конфигурации *nhsrv.ini*.



Протокол NetBIOS может использоваться только с приложениями, защищенными системой HASP4.

### Передача HASP License Manager для Windows пользователям

Если защита вашего программного обеспечения осуществляется с помощью ключа HASP HL Net, вам следует передать конечным пользователям HASP License Manager. Для получения более подробной информации обратитесь также к разделу «Aladdin Monitor».

Существует два способа передачи HASP License Manager конечным пользователям:

- Вы можете передать конечным пользователям установочный файл *lmsetup.exe*. Инструкции по установке программы приводятся в следующем разделе. Инсталляционный файл находится на установочном компакт-диске HASP HL в директории: `windows/Installed/Redistribute/LM/`
- Процесс установки утилиты можно интегрировать с помощью HASP License Manager Installation API в программу установки защищенного приложения. Описание этого API содержится в файле помощи по HASP License Manager.

### Установка HASP License Manager в операционной системе Windows

Оба типа HASP License Manager могут устанавливаться с помощью файла *lmsetup.exe*.

HASP License Manager следует устанавливать на рабочую станцию, к которой подключен ключ HASP HL Net.

Процесс установки можно настроить двумя способами:

- С помощью файла конфигурации *nhsrvw32.ini*.
- С помощью API (только для служб Win32).

### На рабочих станциях с операционной системой Windows 98SE/ME

Файлом HASP License Manager для Windows является *nhsrv.ini*. Для установки License Manager следует использовать файл *lmsetup.exe*.

1. Установите драйвер устройства HASP HL и подключите к компьютеру ключ HASP HL Net.
2. Установите HASP License Manager. Для этого вам надо запустить файл *lmsetup.exe* с установочного компакт-диска HASP HL и следовать инструкциям, появляющимся на экране.

### На рабочих станциях с операционной системой Windows 2000/XP/2003

Файлом HASP License Manager для Windows 2000/XP/2003 является *nhsrvice.exe*. Для его установки следует запустить *lmsetup.exe*.

1. Установите драйвер устройства HASP HL и подключите к компьютеру ключ HASP HL Net.
2. Установите HASP License Manager. Для этого вам надо запустить файл *lmsetup.exe* с установочного компакт-диска HASP HL и следовать инструкциям мастера установки. В качестве типа установки выберите **Service**.



Вы также можете интегрировать установку службы HASP License Manager в установку вашего приложения. Для этого вам понадобятся примеры применения HASP License Manager Installation API, которые находятся в папке по следующему пути:  
Windows/Installed/Samples/LM Install.

## Запуск и прекращение работы HASP License Manager

### Приложение HASP License Manager

Для запуска HASP License Manager вам следует выбрать данное приложение в меню **Start**. HASP License Manager всегда запускается при загрузке любого протокола, а также при подключении ключа HASP HL Net.

Для выхода из приложения следует выбрать **Exit** в основном меню программы.

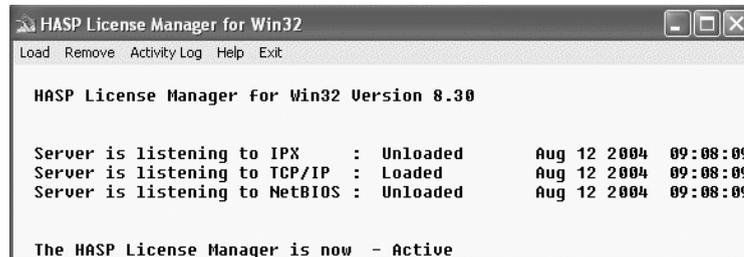
## Служба HASP License Manager

Для запуска и прекращения работы службы HASP License Manager следует использовать раздел **Службы** в Панели Управления. С этой же целью можно применять Aladdin Monitor.

## Работа с HASP License Manager

Работа с HASP License Manager может осуществляться с помощью графического интерфейса. Для его открытия вам следует щелкнуть мышью по его пиктограмме на системной панели.

**Рисунок 12.1** Графический интерфейс HASP License Manager

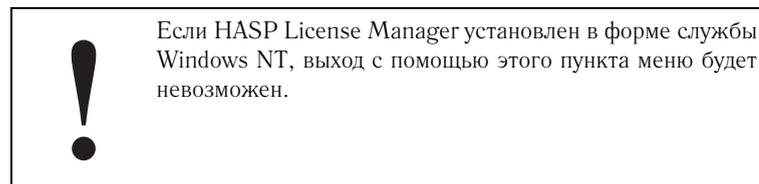


Основное окно HASP License Manager показывает следующую информацию:

- номер версии HASP License Manager;
- Статус каждого протокола (**loaded (загружен)**, **unloaded (не загружен)** или **failed to load (ошибка при загрузке)**) а также дату и время последнего изменения статуса;
- Статус HASP License Manager (активный или неактивный).

Основное окно HASP License Manager можно закрыть, щелкнув мышью по кнопке **Close**. HASP License Manager продолжит работать, а его пиктограмма останется на системной панели.

Для выхода из программы следует выбрать **Exit** в главном меню программы.



## Загрузка протоколов

Выберите желаемый протокол в меню **Load**. Можно загрузить только те протоколы, которые были установлены на вашем компьютере.

## Отключение протокола

Выберите протокол, который вы хотите отключить, в меню **Remove**.

## Просмотр журнала

Для просмотра списка действий HASP License Manager в меню следует выбрать **Activity Log**. Для просмотра журнала по определенному протоколу можно выбрать желаемый протокол из списка.

## Несколько сетевых карт

HASP License Manager привязывает себя к сетевой карте, установленной по умолчанию.

Если вы хотите, чтобы HASP License Manager обслуживал запросы с других сетевых карт компьютера, в настройке сети Windows следует включить **IP Forwarding**.



При использовании HASP License Manager в среде Windows принимайте во внимание, что сетевая карта, установленная по умолчанию, может отличаться в зависимости от версии Windows.

## HASP License Manager для Mac

Клиентам может быть передана версия HASP License Manager для Mac OS X. Обмен данных осуществляется программой с помощью протокола TCP/IP.

В состав HASP License Manager для Mac входит демон и графический интерфейс. Программа может работать как с интерфейсом командной строки, так и с графическим интерфейсом.

### Передача HASP License Manager для Mac конечным пользователям

Все программное обеспечение, необходимое для передачи HASP License Manager конечным пользователям, находится на установочном компакт-диске HASP HL в следующей директории:

MacOS/Installed/Redistribute/LM/

Программное обеспечение можно передать пользователям четырьмя способами:

- инсталляционный образ, который можно загрузить с FTP-сайта;
- инсталляционный пакет, который может являться частью мульти-пакетной установки;
- исходные файлы, которые можно включить в установку одного пакета;
- скрипты, посылаемые конечным пользователям с инструкциями по установке демона и оболочки командной строки.

Для всех четырех методов в указанной выше директории выделены отдельные папки. В каждой папке также находится файл *readme.txt*, содержащий информацию по процедурам установки.

### Запуск и завершение работы HASP License Manager

Для запуска HASP License Manager следует начать работу приложения в меню приложений, а также запустить демон, выбрав **Start Daemon** в окне приложения. Вы также можете запустить демон с помощью инсталляционного скрипта.



Для осуществления автоматической загрузки HASP License Manager вы можете включить опцию **Activate in system startup**.

### Работа с HASP License Manager

Работа с HASP License Manager осуществляется с помощью графического интерфейса, а также в режиме командной строки.

---

HASP License Manager для Mac отображает следующую информацию:

- имя и IP-адрес сервера;
- возможные ключи;
- был ли активирован демон при загрузке системы;
- статус демона.

Также доступны следующие возможности:

- выбор ключа (только в случае, если в данный момент не выполняется процесс демона);
- запуск и прекращение работы демона;
- запуск демона при загрузке системы.

### Указание имени сервера

HASP License Manager может быть присвоено до шести имен сервера. Чтобы присвоить имя сервера:

1. Остановите работу демона, если он был запущен.
2. Активируйте опцию **SRV NAMES**.
3. Введите до шести имен, разделяя их двоеточиями, точками с запятой или пробелами.

Присвоенные имена вступят в силу после запуска демона.



Избегайте использования в именах серверов символов, не являющихся символами ASCII, поскольку их коды в разных системах могут отличаться. Регистр в именах серверов игнорируется. Имена серверов не могут присваиваться работающему демону.

### Указание файла конфигурации

Для настройки работы HASP License Manager вы можете указать файл конфигурации. Для этого вам следует активировать опцию **CFGFILE** и ввести имя файла и путь к нему (см. раздел «Параметры настройки HASP License Manager»).

### Запуск и завершение работы демона

Для запуска и прекращения работы демона следует использовать кнопки окна приложения.

### Автоматический запуск демона

Демон может автоматически запускаться при загрузке системы. Для этого вам надо включить опцию **Activate in system startup**.

## HASP License Manager для Linux

HASP License Manager может работать со всеми версиями Linux. Однако RPM-пакеты имеются только для:

- Red Hat 8 и Red Hat 9;
- SuSe 8.x и SuSe 9.x.

HASP License Manager для Windows представляет собой демон.

## Передача HASP License Manager конечным пользователям

Все программное обеспечение, необходимое для передачи HASP License Manager конечным пользователям, находится на установочном компакт-диске HASP HL в следующей директории:

`macOS/Installed/Redistribute/LM/`

Существует два метода передачи License Manager конечным пользователям:

- RPM-пакет (процесс установки описывается в следующем разделе)
- Скрипты, посылаемые конечным пользователям с инструкциями по установке демона и оболочки командной строки.

Для обоих методов в указанной выше директории выделены отдельные папки. В каждой папке также находится файл *readme.txt*, содержащий информацию по процедурам установки.

---

## Установка HASP License Manager

Если пользователь работает в операционных системах Linux, указанных выше, возможна автоматическая установка с помощью RPM-пакетов.

### Установка HASP License Manager в среде операционной системы SuSE 8.x или 9.x

Используйте следующий RPM-пакет SuSE:

- `rpm -i HASP HLlm-suse-8.30-1.i386.rpm`

### Установка HASP License Manager в среде операционной системы Red Hat 8 или 9

Используйте следующий RPM-пакет Red Hat:

- `rpm -i HASP HLlm-redhat-8.30-1.i386.rpm`

### Установка HASP License Manager в среде других операционных систем Linux

Программа устанавливается вручную.

1. Разверните архив с помощью:  
`tar-xzf [путь/]linuxlm_8_30.tar.gz`  
Будет создана директория `linux-HASP HLlm_8_30`
2. Перейдите в эту директорию и выполните как root:  
`./dinst`

Эта команда устанавливает HASP License Manager и настраивает систему на автоматический запуск демона при загрузке системы.

## Запуск и прекращение работы HASP License Manager

При корректной установке HASP License Manager будет автоматически запускаться при загрузке системы.

Для прекращения работы HASP License Manager необходимо удалить демон. Например, чтобы прекратить работу HASP License Manager, установленного на системе с Linux Red Hat 7.3, следует ввести:

```
Rpm -e HASP HLlm-redhat
```

## Настройка HASP License Manager

При работе с HASP License Manager вам может потребоваться адаптировать программу к конкретной сетевой среде. Для этого можно использовать один из следующих методов:

- Работа с ключами HASP License Manager.
- Использование файла конфигурации *nhsrv.ini*.
- Использование программного интерфейса License Manager Installation API (только для Win32).

## Ключи для работы HASP License Manager

HASP License Manager можно запустить с разнообразными ключами. Такие команды указывают системе HASP HL, какие протоколы использовать, а также как работать с клиентами HASP HL Net.

**Таблица 12.1** Ключи HASP License Manager

Ключ	Описание	Linux	Windows	Mac
-c	Указывает расположение файла конфигурации HASP License Manager	да	нет	да
-help	Показывает перечень доступных ключей	да	нет	да

## Параметры настройки HASP License Manager

Для настройки работы HASP License Manager используется файл конфигурации *nhsrv.ini*. Пример этого файла находится в папке *Redistribute*.

### Последовательность поиска

Вы можете поместить файл *nhsrv.ini* и исполняемый файл HASP License Manager в одну директорию либо в любое другое место в соответствии с заданной последовательностью поиска, описанной в таблице 12.2.

**Таблица 12.2 Последовательность поиска nhsrv.ini**

Операционная система	Последовательность поиска
Windows 98SE/ME	Папка, где расположен исполняемый файл Текущая папка Системная папка Windows Папка Windows Путь
Windows 2000/XP/ Server 2003	Папка, где расположен исполняемый файл Текущая папка Системная папка Windows 32-bit Системная папка Windows 16-bit Папка Windows Путь
Linux	Для использования файла конфигурации необходимо указать к нему путь с помощью ключа <code>-c</code> .
Mac	Для использования файла конфигурации необходимо указать к нему путь с помощью ключа <code>-c</code> .

### Настройки сервера

Параметры сервера задаются не в командной строке, а в файле *nhsrv.ini*. Таким образом, определение параметров службы License Manager упрощается, а все настройки объединяются в одном файле.

На платформах Win32 файл *nhsrv.ini* рекомендуется помещать в ту же папку, где находится исполняемый файл программы. Для платформ с операционными системами Linux и Mac место расположения файла по умолчанию не определено – вы должны самостоятельно задать файл конфигурации. Например:

```
./HASP HLlm -c /etc/nhsrv.ini
```

## Настройки *nhsrv.ini*

Логические ключи могут иметь следующие значения:

**Таблица 12.3** Логические значения для файла *nhsrv.ini*

1	yes (да)	true (верно)	enabled (включено)
0	no (нет)	false (ложно)	disabled (отключено)

## Общие настройки License Manager

Задать настройки для работы HASP License Manager можно в файле конфигурации *nhsrv.ini*. Файл имеет раздел [NHS\_SERVER], предназначенный специально для определения общих настроек. Ниже приводится перечень ключевых слов этого раздела:

<b>Ключевое слово</b>	NHS_IP_USERLIST
Описание	Максимальное число одновременных подключений к License Manager.
Максимальное значение	65520.
Значение по умолчанию	250
Пример	NHS_IP_USERLIST = 1000
<b>Ключевое слово</b>	NHL_IP_SERVERNAMES
Описание	Имя сервера (должно совпадать с именем в запросах клиентов). Максимум 6 имен. Не более 7 знаков на каждое имя. Несколько имен разделяется запятыми.
Значение по умолчанию	Отсутствует
Пример	NHS_IP_SERVERNAMES= cad, 3242e3
<b>Ключевое слово</b>	NHS_IP_HIGHPRIORITY
Описание	Запускает License Manager с высоким приоритетом. По умолчанию программа запускается с обычным приоритетом. Ключ применяется только к операционным системам Win32. Если данный ключ установлен

на «yes», следует проверить, как это повлияло на выполнение других служб.

Значение по умолчанию  
по – запускается с обычным приоритетом  
Пример NHS\_IP\_HIGHPRIORITY= по

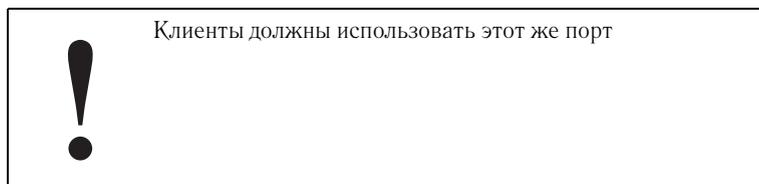
### Настройки IP-протокола

Задать настройки для работы HASP License Manager можно в файле конфигурации *nhsrv.ini*. Файл имеет раздел [NHS\_IP], предназначенный для определения настроек IP-протокола. Ниже приводится перечень ключевых слов этого раздела:

**Ключевое слово** NHS\_USE\_UDP  
Описание Параметр может быть либо включен, либо отключен.  
Значение по умолчанию enabled (включено)  
Пример NHS\_USE\_UDP=enabled

**Ключевое слово** NHS\_USE\_TCP  
Описание Параметр может быть либо включен, либо отключен.  
Значение по умолчанию enabled (включено)  
Пример NHS\_USE\_TCP=enabled

**Ключевое слово** NHS\_IP\_PORTNUM  
Описание Номер IP-порта. Ключ применим только к Win32. Номер порта 475 зарегистрирован за Aladdin License Manager  
Значение по умолчанию 475  
Пример NHS\_IP\_PORTNUM=475



<b>Ключевое слово</b>	NHS_IP_LIMIT
Описание	Определяет диапазон станций, которым разрешено получать доступ к HASP License Manager. Последний байт может означать диапазон. Несколько значений должны разделяться запятой. Перечень может занимать несколько строк. Возможна следующая комбинация универсальных символов (звездочка): четвертый, четвертый, третий или четвертый, третий, второй. Может задаваться дополнительная маска в виде числа простых битов (например, 10.0.0.0/8).
Значение по умолчанию	отсутствует
Пример	NHS_IP_LIMIT = 10.242.18-99,10.1.1.9/16, 10.25.0.0/24,192.0.0*,194.0*,*,11.*,*, *10.24.7.8-12/30,10.24.2.17

### **Настройки протокола IPX (только Win32)**

Задать настройки для работы HASP License Manager можно в файле конфигурации *nhsrv.ini*. Файл имеет раздел [NHS\_IPX], предназначенный для определения настроек протокола IPX. Ниже приводится перечень ключевых слов этого раздела:

<b>Ключевое слово</b>	NHS_USE_IPX
Описание	Настройка может быть либо включена, либо отключена
Значение по умолчанию	enabled (включено)
Пример	NHS_USE_IPX= yes

<b>Ключевое слово</b>	NHS_ADDRPATH
Описание	Путь к файлу <i>HASP Hladdr.dat</i> . IPX-адрес License Manager записывается в этот файл, находящийся в указанной директории. Клиенты могут указать файл в своем файле <i>netHASP.ini</i> .
Значение по умолчанию	текущая директория
Пример	NHS_ADDRPATH=c:\temp

---

<b>Ключевое слово</b>	NHS_APPENDADDR
Описание	Записывает адрес в файл <i>hapaddr.dat</i> . Если настройка включена, текущий адрес License Manager добавляется в <i>HASP Hladdr.dat</i> . Настройка особенно полезна в том случае, когда в сети работает несколько программ License Manager. Клиент может искать все License Manager. При добавлении адреса License Manager не ищет дубликатов файла.
Значение по умолчанию	заменить
Пример	NHS_APPENDADDR=no
<b>Ключевое слово</b>	NHS_USESAP
Описание	Настройка может быть включена, что позволяет показывать HASP License Manager в сети через протокол SAP. Этот протокол позволяет клиентам находить License Manager в различных подсетях. На компьютерах с операционными системами Windows 2000/XP/Server 2003 протокол IPX обычно настроен на создание виртуальных подсетей, поэтому SAP в данном случае необходим.
Значение по умолчанию	enabled (включено)
Пример	NHS_USESAP=enabled
<b>Ключевое слово</b>	NHS_IPX_SOCKETNUM
Описание	Номер сокета IPX. Все клиенты по умолчанию должны использовать этот номер. Номер не следует менять.
	<div style="border: 1px solid black; padding: 10px;"><p>Клиенты должны использовать этот же порт</p></div>
Значение по умолчанию	0x7483
Пример	NHS_IPX_SOCKETNUM= 0x7483

---

## Настройки протокола NetBIOS

Задать настройки для работы HASP License Manager можно в файле конфигурации *nhsrv.ini*. Файл имеет раздел [NHS\_NETBIOS], предназначенный для определения настроек протокола NetBIOS. Ниже приводится перечень ключевых слов этого раздела:

<b>Ключевое слово</b>	NHS_USE_NETBIOS
Описание	Настройка может быть либо включена, либо выключена. Если вы уверены в том, что для обмена данными протокол NetBIOS вам не нужен, выключите эту возможность, что позволит сохранить ресурсы сети и памяти компьютера.
Значение по умолчанию	enabled (включено)
Пример	NHS_USE_NETBIOS= yes
<b>Ключевое слово</b>	NHS_NBNAME
Описание	Применяется для изменения имени. Клиенты должны использовать то имя, которое не используется в вашем пространстве имен NetBIOS. Имена должны следовать за условными обозначениями NetBIOS
Значение по умолчанию	enabled (включено)
Пример	NHS_NBNAME=MyNBName
	<div style="border: 1px solid black; padding: 5px;"> Эту настройку следует использовать только в том случае, если вы уверены в необходимости указать новое имя NetBIOS.</div>
<b>Ключевое слово</b>	NHS_USE-LUNA_NUMS
Описание	Предназначено для обеспечения совместимости со старыми версиями.
Значение по умолчанию	all (автоматически)
Пример	NHS_USE_LUNA_NUMS=3, 0, 7, 2

---

## Настройка клиента HASP HL Net

В данном разделе описывается процесс настройки клиента HASP HL Net (приложения, защищенного с помощью ключа HASP HL Net) с применением файла конфигурации.

Когда клиент находит файл конфигурации, информация и значения считываются из файла. Если файл конфигурации не был найден, используются значения по умолчанию.

В файле конфигурации клиента HASP HL Net вы можете настроить процесс поиска клиентом HASP License Manager.

По умолчанию, файл конфигурации называется *nethasp.ini*. Копия этого файла находится в директории, в которую была установлена программа HASP License Manager. Если вы изменяете имя файла конфигурации, это следует отразить при реализации защиты с помощью HASP HL Envelope или HASP HL API.

### Последовательность поиска файла конфигурации

Последовательность поиска файла конфигурации зависит от операционной системы и типа приложения.

Защищаемое приложение осуществляет поиск файла конфигурации после первого вызова на открытие сессии (см. раздел «Функция login в HASP HL API»). Последовательность поиска описана в таблице 12.4.

Таблица 12.4

Операционная система	Последовательность поиска
Win32	Директория, где находится исполняемый файл Текущая директория Системная папка Windows Папка Windows Путь
Mac OS X (Carbon)	Текущая директория
Mac OS X	Текущая директория Директория текущего пользователя Прочие директории



В системах, работающих под управлением MAC OS X, поиск файла *nethasp.ini* осуществляется без начального периода. Если вы используете систему Mac OS X с учетом регистра, убедитесь в том, что имя файла *nethasp.ini* указано в нижнем регистре.

## Разделы файла конфигурации

Файл конфигурации клиента HASP HL Net содержит три дополнительных раздела:

- [NH\_COMMON] — для общих настроек
- [NH\_IPX] — для настроек протокола IPX
- [NH\_TCP/IP] — для настроек протокола TCP/IP

Раздел [NH\_COMMON] предназначен для указания общих настроек для всех разделов файла конфигурации. В остальных двух разделах настройки определяют параметры работы конкретного протокола.

## Определение ключевых слов

В каждом разделе вы можете задать значение, как общих ключевых слов, так и ключевых слов раздела. Если вы задаете значение общего ключевого слова в разделе, предназначенном для определения параметров протоколов, то данное значение для данного протокола используется вместо значения ключевого слова, заданного в разделе [NH\_COMMON].

Для настройки дополнительных параметров определенных протоколов следует использовать ключевые слова, предназначенные для раздела.

Настройки API и Envelope имеют приоритет перед настройками файла конфигурации.

Каждая строка файла конфигурации, полученного с программным обеспечением HASP HL, начинается с точки с запятой (;). Для применения параметров, заданных в строке, следует убрать этот знак. Если вы хотите добавить комментарии, перед ними следует ставить точку с запятой.



В именах файлов конфигурации и ключевых словах регистр не имеет значения. Исключение составляет случай использования чувствительной к регистру системы Mac OS X.

Ниже описываются разделы файла конфигурации клиента HASP HL Net. Для каждого слова приводятся возможные значения и короткое описание. Также приводятся значения по умолчанию (если они есть).

## [NH\_COMMON]

### [NH\_COMMON] – ключевые слова для разделов

<b>Ключевое слово</b>	nh_ipx
Описание	Разрешает использование протокола IPX
Возможное значение	включено, отключено
<b>Ключевое слово</b>	nh_tcpip
Описание	Разрешает использование протокола TCP/IP
Возможное значение	enabled, disabled (включено, отключено)

### [NH\_COMMON] – общие ключевые слова

<b>Ключевое слово</b>	nh_session
Описание	Устанавливает максимальный период времени, в течение которого приложение пытается совершить обмен данными с HASP License Manager.
Возможное значение	числовое значение
По умолчанию	2 секунды

<b>Ключевое слово</b>	nh_send_rcv
Описание	Устанавливает максимальную продолжительность времени, в течение которого HASP License Manager пытается послать или получить пакет.
Возможное значение	числовое значение
По умолчанию	1 секунда

## [NH\_IPX]

### [NH\_IPX] – ключевые слова для разделов

<b>Ключевое слово</b>	nh_use_bindery
Описание	Позволяет протоколу IPX работать с Novell BINDERY.
Возможное значение	enabled, disabled (включено, отключено)
По умолчанию	Disabled
<b>Ключевое слово</b>	nh_use_broadcast
Описание	Позволяет использовать широковещательный механизм IPX
Возможное значение	enabled, disabled (включено, отключено)
По умолчанию	Enabled
<b>Ключевое слово</b>	nh_bc_socket_num
Описание	Указывает номер сокета для широковещательного механизма. Число в шестнадцатиричном формате
Возможное значение	Значение в шестнадцатиричном формате
По умолчанию	7483H
<b>Ключевое слово</b>	nh_use_int
Описание	Значение 2F_NEW означает, что протокол IPX использует только прерывание 2Fh. Значение 7F_OLD означает, что протокол IPX использует только прерывание 7Ah.
Возможное значение	2F_NEW, 7F_OLD
По умолчанию	2F_NEW

---

<b>Ключевое слово</b>	nh_server_name
Описание	Обмен данными с HASP License Manager с указанным именем.
Возможное значение	<имя1>, <имя2>
<b>Ключевое слово</b>	nh_search_method
Описание	Задаёт метод обмена данными. Обмен данным защищенного приложения с HASP License Manager может происходить либо в сетевой среде, либо через Интернет.
Возможное значение	localnet, internet (локальная сеть, Интернет)
По умолчанию	Localnet
<b>Ключевое слово</b>	nh_datfile_path
Описание	Задаёт расположение адресного файла HASP License Manager

### **[NH\_IPX] – общие ключевые слова**

<b>Ключевое слово</b>	nh_session
Описание	Устанавливает максимальный период времени, в течение которого приложение пытается совершить обмен данными с HASP License Manager.
Возможное значение	числовое значение
По умолчанию	2 секунды
<b>Ключевое слово</b>	nh_send_rcv
Описание	Устанавливает максимальную продолжительность времени, в течение которого HASP License Manager пытается послать или получить пакет.
Возможное значение	числовое значение
По умолчанию	1 секунда

### **[NH\_TCP] – ключевые слова для разделов**

<b>Ключевое слово</b>	nh_server_addr
Описание	Указывает IP-адреса всех HASP License Manager. Количество строк и число адресов не ограничено.
Возможное значение	<адрес1>, <адрес2>

---

Примеры	IP addresses: 192.114.176.65 Local hostname: ftp.aladdin.com
<b>Ключевое слово</b>	nh_session
Описание	Устанавливает максимальный период времени, в течение которого приложение пытается совершить обмен данными с HASP License Manager.
Возможное значение	числовое значение
По умолчанию	2 секунды
<b>Ключевое слово</b>	nh_send_rcv
Описание	Устанавливает максимальную продолжительность времени, в течение которого HASP License Manager пытается послать или получить пакет.
Возможное значение	числовое значение
По умолчанию	1 секунда

## Настройка среды HASP HL Net

В разделе рассматриваются дополнительные настройки HASP License Manager и ключевые слова для файла *nethasp.ini*, призванные обеспечить работу ключа HASP HL Net в конкретной сетевой среде.

## Определение перечня станций при применении протокола IPX

При использовании протокола IPX вы можете определить станции различных сегментов сети, которые будут получать доступ к HASP License Manager.

Чтобы разрешить доступ из различных сегментов сети:

1. Загрузите HASP License Manager с ключом **-ipxnosap**. Применение этого ключа гарантирует, что адрес HASP License Manager показывается в сети не через механизм протокола SAP, а с помощью адресного файла *newhaddr.dat*.
2. Отредактируйте файл *nethasp.ini* следующим образом:
  - В разделе [NH\_COMMON] укажите NH\_IPX=Enabled
  - В разделе [NH\_IPX] укажите NH\_USE\_BROADCAST=Disabled
  - В разделе [NH\_IPX] укажите NH\_USE\_BINDERY=DisabledЭти настройки укажут защищенным приложениям, работающим

---

на указанных станциях, искать адресный файл и считывать из него адрес HASP License Manager.

3. Скопируйте защищенное приложение и файл *nethasp.ini* в одну директорию.

## Определение перечня станций при использовании протокола TCP/IP

При использовании протокола TCP/IP существует два способа определения перечня станций. Вы можете либо указать станции, обслуживаемые HASP License Manager, либо указать выбранным станциям искать определенную программу HASP License Manager.

### Определение станций с помощью *nhsrv.ini*

HASP License Manager для Windows, Win32 и Mac могут обращаться к файлу конфигурации *nhsrv.ini*. В этом файле можно указать те станции, которые будут обслуживаться HASP License Manager.

#### Чтобы определить станции:

Отредактируйте файл *nhsrv.ini* следующим образом:

- В разделе [NHS\_SERVER] укажите  
NHS\_IP\_LIMIT=<ipaddr>[, ipaddr...]

#### Примеры форматов <ipaddr>

Для определения станций с помощью файла *nhsrv.ini* вы можете использовать один из следующих форматов:

- 10.1.2.3

HASP License Manager будет обслуживать только станцию с указанным адресом.

- 10.1.2.\*

HASP License Manager будет обслуживать только станции, соответствующие указанной маске IP-адресов, т.е. с 10.1.2.0 по 10.1.2.255.

- 10.1.\*.\*

HASP License Manager будет обслуживать только станции, соответствующие указанной маске IP-адресов, т.е. с 10.1.0.0 по 10.1.255.255.

#### Ограничение доступа станций к ключу HASP HL Net в сети TCP/IP

1. Задайте в файле *nhsrv.ini* маску адресов для станций.

2. Скопируйте файл *nhsrv.ini* в место, доступное HASP License Manager.

### **Определение станций с помощью *nethasp.ini***

Вы можете указать в файле конфигурации HASP HL Net, что HASP License Manager (в соответствии с адресом) будет искать только заданные станции.

#### **Чтобы определить станции:**

1. Отредактируйте файл *nethasp.ini*: в разделе [NH\_TCPIP] укажите NH\_SERVER\_ADDRESS=<адрес HASP License Manager>
2. Скопируйте файл *nethasp.ini* в место, доступное только желаемым станциям.

### **Настройка времени ожидания**

HASP License Manager не может одновременно обрабатывать больше одного запроса. Время ожидания определяет, сколько времени защищенное приложение будет пытаться получить доступ к HASP License Manager.

Практически во всех сетях время ожидания достаточное. В сетях, где ключ HASP HL Net подключен к медленной или загруженной станции, указанное по умолчанию значение придется изменить.

#### **Чтобы задать время ожидания:**

В соответствующем разделе файла *nethasp.ini* укажите:

NH\_SESSION = <m>

NH\_SEND\_RCV = <n>

Значения m и n указываются в секундах. По умолчанию, m = 2 секунды, n = 1 секунда.

## Определение числа обслуживаемых защищенных приложений

HASP License Manager позволяет изменить число обслуживаемых станций, указанное по умолчанию (250 для Win32, Mac и Linux).

HASP License Manager распределяет память в зависимости от максимального числа защищенных приложений. При необходимости вы можете изменить значение по умолчанию, что позволит сберечь память.

### Чтобы изменить распределение памяти

Загрузите HASP License Manager со следующим ключом:

```
nhsrvw32 -userlist = n
```

Значение *n* — число обслуживаемых станций. Ключ **userlist** доступен только для Win32.



# Aladdin Monitor

В главе рассматриваются следующие темы, описывающие работу утилиты Aladdin Monitor:

- установка Aladdin Monitor;
- настройки утилиты Aladdin Monitor;
- мониторинг работы HASP License Manager;
- проверка ключей HASP HL Net.

Утилита Aladdin Monitor позволяет осуществлять централизованное администрирование приложений HASP License Manager и ключей HASP HL Net.

Aladdin Monitor может работать в среде следующих операционных систем: Windows 98SE/ME/NT/2000/XP/Server 2003. Обмен данным происходит с помощью протоколов TCP/IP и IPX.

Утилиту Aladdin Monitor следует передавать конечным пользователям вместе с:

- Защищенным программным обеспечением
- Ключами HASP HL Net
- HASP License Manager

Aladdin Monitor имеет встроенную справочную систему.



Работу Aladdin Monitor можно настроить с помощью файла конфигурации клиента HASP HL Net (*nethasp.ini*). Подробная информация по этому вопросу содержится в разделе «Настройка клиента HASP HL Net».

Утилита позволяет:

- проверять свойства HASP License Manager;

- проверять ключи HASP HL Net;
- запускать и прекращать работу службы HASP License Manager.

## Установка Aladdin Monitor

Утилиту Aladdin Monitor можно установить на любой рабочей станции сети. Устанавливать на этой же станции HASP License Manager необязательно.

Для установки программы следует использовать установочную утилиту *aksmon32.exe*. После запуска установочной утилиты следуйте появляющимся на экране инструкциям.

## Настройка Aladdin Monitor

Вы можете изменить следующие установки Aladdin Monitor:

- Язык интерфейса — английский или немецкий
- Частоту обновления диалогового окна — значение по умолчанию составляет 2 секунды
- Частоту сетевых запросов — значение по умолчанию составляет 3 минуты
- Режим работы Aladdin Monitor — режим HASP, Hardlock или оба режима вместе.
- Использовать или не использовать файл конфигурации *nethasp.ini*

Для изменения настроек следует выбрать пункт **Settings** в меню **File**. Изменения вступают в силу после перезапуска программы.

## Мониторинг работы HASP License Manager

В левой части окна **Aladdin Network Resources** выберите программу HASP License Manager, работу которой вы хотите проверить.



Если в указанной области окна не показывается ни одна программа HASP License Manager, дважды щелкните по папке **HASP License Manager** или обновите содержимое окна, выбрав **File/Rescan**.

В правой части окна отобразится информация о HASP License Manager.



Программы HASP License Manager, осуществляющие обмен данными только с помощью протокола NetBIOS, не распознаются Aladdin Monitor.

О выбранной программе HASP License Manager отображается следующая информация:

- общая информация о выбранной программе HASP License Manager (таблица 13.1);
- информация об управляемых ключах HASP HL Net (таблица 13.2).

**Таблица 13.1 Информация о HASP License Manager**

Поле	Описание
Name (Имя)	Имя компьютера, на котором работает HASP License Manager.
Version (Версия)	Версия HASP License Manager
IP	IP-адрес компьютера
IPX	IPX-адрес компьютера
LM Type (Тип License Manager)	Платформа, на работу с которой настроена программа HASP License Manager
TCP/IP, IPX	Используемые в настоящий момент протоколы

**Таблица 13.2 Информация о ключе HASP**

Поле	Описание
HASP # (Номер HASP)	Идентификатор сетевого ключа HASP
Key Type (Тип ключа)	Тип ключа HASP: HASP или более старый, HASP4 или HASP HL
Key Model (Модель ключа)	Максимальное число лицензий
Current Stations (Текущие станции)	Подключенные в настоящий момент станции

## Проверка ключей HASP HL

В левой части окна под папкой HASP License Manager выберите ключ HASP HL. Ключ HASP HL можно проверить только в том случае, если была открыта сессия.



Если ключ не отображается, дважды щелкните по значку соответствующей программы HASP License Manager, либо обновите окно, выбрав **File/Rescan**.

В правой части окна будет отображаться следующая информация о ключе:

- общая информация о ключе HASP HL (таблица 13.3);
- обзор программ (таблица 13.4);
- обзор сессий по отдельным программам (таблица 13.5).

**Таблица 13.3 Информация о ключе HASP HL**

Поле	Описание
HASP # (Количество HASP)	Общее число ключей HASP
Key Type (Тип ключа)	Тип ключа HASP: HASP HL, HASP или HASP4

**Таблица 13.4 Таблица программ**

Поле	Описание
Program No. (Номер программы)	Номер, представляющий защищенную программу.
Current Stations (Текущие станции)	Подключенные в данный момент станции.
Maximum Stations (Станций максимально)	Максимально возможное количество станций.
Activations (Активации)	Максимальное количество запусков программы

**Таблица 13.5 Таблица открытых сессий**

<b>Поле</b>	<b>Описание</b>
No. (Количество)	Общее количество открытых сессий.
Login ID (Идентификатор открытой сессии)	Адрес станции, по которому проходило открытие сессии.
Host Name (Имя компьютера)	Имя компьютера с подключенным ключом.
Protocol (Протокол)	Используемый протокол.
Timeout (Время ожидания)	Период времени бездействия, после которого открытая сессия удаляется (в секундах)

## Служба HASP License Manager

Служба HASP License Manager позволяет осуществлять администрирование ключей HASP HL Net на рабочих станциях Windows NT. Для запуска и прекращения работы службы на локальном компьютере можно использовать Aladdin Monitor.

### Запуск службы

В меню **HASP LM Service** или **Services/HASP** выберите **Start HASP LM Service**. После этого служба запускается, а подключенные к локальному компьютеру ключи HASP HL Net становятся доступными в сети.

Запустить службу также можно с помощью контекстного меню. Для этого вам следует щелкнуть правой кнопкой мыши на папке **HASP License Manager** и выбрать **Start HASP LM**.

### Прекращение работы службы

В меню **HASP LM Service** или **Services/HASP** выберите **Stop HASP LM Service**. Работа службы прекращается, затем обновляется информация окна. Это может занять некоторое время, поскольку происходит сканирование всей сети.

Прекратить работу службы также можно с помощью контекстного меню. Для этого вам следует щелкнуть правой кнопкой мыши на папке **HASP License Manager** и выбрать **Stop HASP LM**.

# Диагностика ключей HASP HL

В главе описывается применение программ Aladdin DiagnostiX и Aladdin DiagnostiX Memory Beamer в целях сбора информации о ключах HASP HL. Рассматриваются следующие темы:

- передача Aladdin DiagnostiX клиентам;
- диагностика ключей HASP HL;
- создание отчетов по ключам HASP HL;
- интеграция Aladdin DiagnostiX и внешней программы для создания отчетов;
- применение Aladdin DiagnostiX Memory Beamer.

Утилита Aladdin DiagnostiX собирает информацию о ключах HASP HL и системах, к которым эти ключи подключены. Подобный механизм обратной связи позволяет предоставить клиентам решение проблем, с которыми они сталкиваются при использовании защищенного программного обеспечения.

## Функции диагностики утилиты Aladdin DiagnostiX

Утилита Aladdin DiagnostiX может применяться для:

- проверки ключа HASP HL;
- создания файла отчета, содержащего информацию об аппаратных устройствах Aladdin и прочую важную информацию о системе;
- загружать при необходимости обновленный драйвер HASP HL.

Программа Aladdin DiagnostiX работает в средах следующих операционных систем: Windows 98SE/ME и Windows NT/2000/XP/Server 2003.

Когда ваши клиенты запускают Aladdin DiagnostiX, для проверки ключа HASP HL им следует выбрать инструмент **Check HASP HL**. В дополнение к информации о ключе HASP HL утилита позволяет генерировать отчеты,

содержащие основную информацию о вашей системе. Эти отчеты передаются службе технической поддержки для дальнейшего анализа.

При запуске Aladdin DiagnostiX происходит проверка системы с помощью инструмента **System Info**. Результаты проверки отображаются в рабочем поле программы как часть экрана **System Info**.

## Передача Aladdin DiagnostiX пользователям

Aladdin DiagnostiX следует передать клиентам. Установка программы осуществляется просто — пользователь должен запустить файл *aksdiag32.exe* и следовать появляющимся на экране инструкциям.

Копия этого файла находится в папке **Redistribute** установочного компакт-диска HASP HL. Его следует переслать пользователям по электронной почте или записать на дистрибуционный диск вашего защищенного приложения.



Для получения доступа и обмена данными с ключом HASP HL необходим код разработчика. В целях сохранения секретности мы рекомендуем передавать клиентам библиотеку DLL, в которой этот код содержится в зашифрованном виде.

## Диагностика ключей HASP HL

В состав утилиты Aladdin DiagnostiX входит инструмент **Check HASP HL**, позволяющий получать доступ к ключам HASP HL и получать с них необходимую информацию.

### Проверка ключа HASP HL

Чтобы проверить ключ HASP HL, на вкладке **Diagnostic Tools** следует щелкнуть по значку **HASP HL**. В рабочем поле откроется окно **Check HASP HL**.

### Процедура проверки ключа HASP HL

Проверка ключа HASP HL осуществляется в несколько шагов.

1. На экране **Check HASP HL** выберите **Search Mode**. Значение по умолчанию — **Local and Remote**.
2. Укажите для ключа Код разработчика.
3. Укажите номер программы. Мы рекомендуем указывать (0), поскольку, если ключ HASP HL подключен к компьютеру, эта функция всегда доступна.
4. Если вы хотите сохранить содержимое памяти ключа, отметьте опцию **Save update information when checking key**.

5. Щелкните по кнопке **Check Key**.
6. Информация о получении доступа к ключу будет показана на вкладке **Key Access History** программы Aladdin DiagnostiX.

## **Вкладка Key Access History**

Вкладка **Key Access History** регистрирует все попытки получения доступа к ключам HASP HL. Информация о доступе к ключу показывается последовательно, в первую очередь выводятся сведения о самых последних попытках получения доступа. В таблице 14.1 описывается информация, показываемая на данной вкладке.

**Таблица 14.1 Вкладка Key Access History утилиты Aladdin DiagnostiX**

Столбец	Значение	Описание
Access Mode (Режим доступа)	local (локальный)	Ключ был найден на локальном порту.
	remote (удаленный)	Ключ был найден на удаленном порту
Prog No. (Номер программы)	Номер программы, примененной для открытия сессии с ключом HASP HL.	
Port/IP	Показывает номер порта, к которому подключен ключ HASP HL. Если ключ был подключен, и к нему был получен доступ через удаленный порт с помощью HASP License Manager, показывается IP-адрес. Если к ключу HASP HL невозможно получить доступ, показывается <b>n/a</b> .	
Key Type (Тип ключа)	Тип обнаруженного ключа HASP HL.	
HASP ID (Идентификатор ключа)	ID-номер обнаруженного ключа.	
Addit. Info (Дополнительная информация)	Дополнительная информация, связанная с показанным при работе кодом ошибки	

### Настройка файла *nethasp.ini*

Утилита Aladdin DiagnostiX позволяет клиентам быстро настраивать и создавать файлы *nethasp.ini*, что повышает скорость обмена данными между защищенными приложениями и ключами HASP HL Net.

Чтобы открыть экран **nethasp.ini Configuration**:

1. на вкладке **Diagnostic Tools** дважды щелкните по значку **HASP**;
2. щелкните по значку **HASP nethasp.ini**.

В рабочей области откроется экран **nethasp.ini Configuration**.

Ваши клиенты могут создать файл *nethasp.ini* тремя способами:

- Создать файл автоматически (**automated**). Файл будет создан на основе информации, собранной программой при загрузке. В дальнейшем содержащаяся в файле информация будет использоваться для получения доступа к серверу. Выбор протокола осуществляется в следующей последовательности: IP, затем IPX. При создании файла могут использоваться только имена серверов, найденные при запуске программы.
- Создать файл по умолчанию (**default**). В этом случае клиент может задать в *.ini*-файле использование IP или IPX. Если для создания применяется этот вариант, в файл не будут включаться какие-либо заданные заранее имена серверов.
- Создать файл, настроенный пользователем (**customized**). Программа Aladdin DiagnostiX позволяет задать определенный локальный протокол и протокол сервера.

### Настройка файлов *nethasp.ini*

Чтобы настроить *.ini*-файл с помощью Aladdin DiagnostiX:

1. Выберите один из трех методов создания файла. Если вы выбрали метод **automated**, шаги 2-3 следует пропустить и перейти сразу к шагу 4.
2. Если в рамках первого шага вы выбрали метод **default**, выберите один или оба протокола и перейдите к шагу 4.
3. Если вы выбрали метод **customized**, используйте специальные поля для задания IP- и IPX-серверов. Для этого вы можете использовать параметры командной строки или щелкнуть по соответствующей кнопке обзора и сделать выбор из имеющихся возможностей.
4. В поле **Set output directory** введите имя и укажите директорию, куда будет помещен созданный и настроенный файл. Для этого также можно использовать кнопку обзора.
5. Щелкните по кнопке **Create**.

## Создание отчетов о ключах HASP HL

Ваши клиенты могут применять Aladdin DiagnostiX для создания отчетов, содержащих:

- информацию о ключах HASP HL;
- информацию о системе;
- информацию из внешнего инструмента создания отчетов;
- содержимое памяти ключей HASP HL.

Если у клиентов возникают проблемы с ключами HASP HL или при получении доступа к защищенному приложению, они могут отправить созданные отчеты специалистам технической поддержки Aladdin.

### Настройки, используемые при создании отчета

Содержание и формат отчетов, создаваемых Aladdin DiagnostiX, определяются в специальном окне настроек. Для задания настроек необходимо открыть вкладку **Create Report**.

Настройки, имеющиеся на вкладке **Create Report** включают:

**Report format:** позволяет выбрать один из трех возможных форматов — XML, HTML или TEXT.



Если вы выбираете формат HTML или TEXT, будет также создан дополнительный XML-файл, в котором будет находиться содержимое памяти обнаруженного ключа.

Утилита также позволяет генерировать несколько файлов. Чтобы заархивировать все сгенерированные файлы отметьте опцию **Zip all output files**.

Чтобы добавить в файл информацию по подсистемам Win16 и DOS, следует отметить опцию **Include Win16/DOS**. Aladdin DiagnostiX создаст отдельный файл отчета для каждой подсистемы.

### Чтобы создать отчет

1. Измените все необходимые настройки, описанные в разделе выше.
2. Создайте отчет с помощью одного из следующих методов:
  - щелкните мышью по значку **Create Report**;
  - в меню **Edit** выберите **Create Report**;
  - нажмите <Ctrl> + R;
  - щелкните по кнопке **Create Report** на экране **System Info**.

Появится окно сообщения, информирующее вас о содержимом файла и его расположении.

## Интеграция с внешним инструментом генерирования отчетов

Aladdin DiagnostiX позволяет задать настройки для двух различных инструментов генерирования отчетов, аналогичных msinfo32. Созданные внешними инструментами отчеты могут архивироваться и храниться в виде отдельного файла. Для создания связи между Aladdin DiagnostiX и

---

внешним инструментом создания отчетов следует применить следующую процедуру:

1. В окне **Aladdin DiagnostiX Settings** откройте вкладку **Reporting Tools**.
2. Отметьте **Reporting tool 1** – будут активированы поля **Execute** и **Output File**.
3. В поле **Execute** вы можете либо вручную указать путь к инструменту создания отчетов, либо использовать для этого кнопку **browse**. Также можно добавить параметры запуска инструмента и указать результирующий файл Aladdin DiagnostiX.
4. Укажите имя генерируемого внешним инструментом файла в поле **Output File**. Задайте путь к директории, куда файл будет помещен.
5. Для задания настроек для второго внешнего инструмента следует отметить **Reporting Tool 2** и повторить шаги 2-4. У вас есть возможность задать максимальный период времени, в течение которого будет генерироваться отчет. Для этого нужно отметить соответствующую опцию и указать лимит времени. Если внешний инструмент не создаст отчет в указанное время, его работа будет остановлена.
6. Для выхода щелкните по кнопке **OK**.

## Обновление драйверов HASP HL

Aladdin DiagnostiX может использоваться клиентами в целях обновления драйверов подключенных локально ключей. Драйверы доступны для загрузки с ftp-сайта Aladdin.

Чтобы обновить драйверы:

- a. в окне **System Info** щелкните по кнопке **Driver Update**;
- b. для загрузки и установки новых драйверов следуйте появляющимся на экране инструкциям.

Обновленные драйверы будут немедленно отображены в окне **System Info**.

## Aladdin DiagnostiX Memory Beamer

Aladdin DiagnostiX Memory Beamer является утилитой разработчика, предназначенной для передачи данных между разработчиком и клиентом. В связке с Aladdin DiagnostiX утилита Aladdin DiagnostiX Memory Beamer позволяет:

- Пересылать пользователям настроенные библиотеки DLL.
- Получать доступ к отчетам, созданным с помощью Aladdin DiagnostiX.
- Считывать информацию об обновлениях, включая содержимое памяти переданных пользователям ключей HASP HL. Информация может сохраняться в формате C2V-файла, что позволяет обрабатывать ее с помощью HASP HL Factory.



Утилита Aladdin DiagnostiX Memory Beamer предназначена для использования исключительно разработчиком программного обеспечения, а не пользователями.

### Пересылка пользователям настроенных библиотек DLL

Утилиту Aladdin DiagnostiX Memory Beamer можно использовать для вставки кодов разработчика в библиотеки DLL, которые в дальнейшем передаются пользователям. Такие библиотеки после переноса на системы пользователей являются дополнительным средством получения доступа к ключу HASP HL с помощью Aladdin DiagnostiX.

#### Для подготовки библиотеки DLL:

1. Задайте путь к файлу Кода разработчика.
2. Щелкните по кнопке **Inject**.  
Появится сообщение, подтверждающее, что код был вставлен в библиотеку. Файл библиотеки (*custom.dll*) должен находиться в текущей директории. Не меняйте его имени!
3. Перешлите библиотеку пользователю. Проинструктируйте его поместить полученный файл в ту же папку, где находится утилита Aladdin DiagnostiX. Пользователь может также поместить полученную библиотеку в папку, содержащую инсталлятор Aladdin DiagnostiX. В этом случае инсталлятор автоматически скопирует библиотеку в нужную директорию.

---

## Чтение файлов отчетов

Aladdin DiagnostiX Memory Beamer позволяет быстро получать доступ и сортировать отчеты, созданные утилитой Aladdin DiagnostiX. Aladdin DiagnostiX Memory Beamer в соответствии с информацией, хранящейся в файле отчета, выводит следующие данные:

- Тип ключа HASP HL
- Доступ – как осуществляется доступ к ключу.

### Чтобы прочесть файл отчета:

1. Укажите путь к файлу отчета и щелкните по кнопке **OK**.
2. Если в отчете содержатся соответствующие записи о ключах HASP HL, в полях **Key Type and Access** появятся записи.



Если вы хотите пропустить определенный файл отчета и перейти к следующему файлу, щелкните по кнопке **Skip**.

3. Если файл отчета содержит сведения о памяти ключа HASP HL, эту информацию можно сохранить в формате C2V-файла для дальнейшей обработки с помощью HASP HL Factory. Укажите имя файла или используйте кнопку **browse** для задания директории. По умолчанию именем файла является *[haspid].c2v*, где *haspid* является серийным номером ключа HASP HL.



# Устранение неисправностей

В первой части приложения приводится список контрольных вопросов, который поможет вам при необходимости решить простейшие проблемы, с которыми вы можете столкнуться при работе с системой HASP HL. Во второй части рассматриваются конкретные проблемы, с которыми могут столкнуться ваши клиенты или вы, а также приводятся возможные способы их решения.

Ключи HASP HL соответствуют самым высоким стандартам качества. Однако, как и любое другое периферийное устройство для персонального компьютера, ключ, в силу некорректной работы аппаратного обеспечения или неправильной установки, может не работать на некоторых компьютерах. В таких ситуациях может помочь данное приложение.

Для дополнения дополнительной информации, клиенты могут обратиться к материалам Интернет-сайта Aladdin Knowledge Base:

<http://www.hasp.com/kb.2>

Этот ресурс содержит обширный перечень возможных проблем и путей их решения. Чтобы избежать трудностей, убедитесь в том, что вы используете последнюю версию программного обеспечения HASP HL. Получить последние обновления программного обеспечения можно, связавшись с ближайшим представителем Aladdin, или со следующей страницы:

<http://www.hasp.com/download>

Если вас продолжают преследовать проблемы, проверьте правильность работы примеров HASP HL и утилиты Aladdin DiagnostiX. Перешлите результаты проверки ближайшему представителю Aladdin.

## Список контрольных вопросов

Если клиент сообщает о проблеме, в первую очередь следует предпринять ряд действий и выяснить у клиента подробности о возникшей проблеме:

- Обратите внимание на код ошибки и сообщение. Информация о них приводится в разделе «Коды статуса API».
- Правильно ли подключен ключ к порту USB?
- Испытывает ли компьютер клиента такие проблемы, как конфликт устройств, системные ошибки, ошибки при загрузке и т.д.?
- Может ли утилита Aladdin DiagnostiX получить доступ к ключу HASP HL? Постарайтесь создать отчет.
- Случается ли аналогичная проблема, если защищенное приложение запускается на другом компьютере такой же модели?

---

## Проблемы и решения

<b>Проблема</b>	Не устанавливаются драйверы HASP HL.
<b>Решение</b>	Установлены ли на машине более старые версии драйверов устройств HASP? Старые драйверы необходимо удалить с помощью их программы установки. После удаления старых драйверов установите драйверы HASP HL. Для получения информации по процедуре установки вы можете обратиться к разделу «Установка драйверов устройств HASP HL»
<b>Проблема</b>	Для установки драйвера устройств в среде Windows 2000/XP/2003 вы пытаетесь использовать файл <i>haspdinst.exe</i> , но получаете сообщение об ошибке.
<b>Решение</b>	Просмотрите инструкции по установке <i>haspdinst.exe</i> (см. раздел « <i>haspdinst.exe</i> »). Вы также можете попробовать установить драйверы с помощью <i>HASPUUserSetup.exe</i> (см. раздел «Приложение HASPUUserSetup.exe»).
<b>Проблема</b>	Защищенное приложение не может обнаружить ключ HASP HL.
<b>Решение</b>	<p>Горит ли индикатор на самом ключе? Если нет, то неисправность может иметь несколько причин:</p> <ol style="list-style-type: none"><li>1. Ключ подключен к порту USB неправильно. Отключите его, а через несколько секунд вновь подключите. Если индикатор загорится, приложение сможет получить к ключу доступ.</li><li>2. Необходимые драйверы не установлены. Если вы работаете с защищенным приложением в среде Windows, проверьте наличие соответствующей строки в Диспетчере устройств. Если строки, связанной с ключами HASP HL в диспетчере устройств нет, вам следует установить драйверы (см. раздел «Установка драйверов устройств HASP HL»).</li><li>3. Проверьте правильность работы порта USB. Отключите все USB-устройства от портов. Подключите ключ к другому порту USB. Попробуйте подключить другое USB-устройство на том порту, где ранее был подключен ключ HASP HL.</li></ol>

**Проблема** Ваше приложение очень долго ищет ключ HASP HL Net в большой сети, использующей протокол TCP/IP.

**Решение** Рекомендуется настроить механизм поиска. Для указания IP-адреса HASP License Manager, а также выбора метода поиска UDP или TCP можно использовать файл конфигурации HASP HL Net. Применение файла конфигурации приведет к тому, что клиент HASP HL Net будет искать HASP License Manager с указанным IP-адресом, что существенно ускорит процесс поиска (см. раздел «Настройка клиента HASP HL Net».)

---

**Проблема** Во время обмена данными HASP HL Net и HASP License Manager на экране появляется сообщение об ошибке, сообщающее о перегрузке сети.

**Решение** Увеличьте время ожидания клиента HASP HL Net (см. раздел «Настройка клиента HASP HL Net».)

---

**Проблема** Вы получаете сообщение об ошибке, говорящее, что утилита HASP License Manager не найдена.

**Решение** Это сообщение может появляться при работе с протоколами TCP/IP/IPX, если вы используете широковещательный механизм поиска. Увеличьте время ожидания в файле *nethasp.ini* до 8 секунд (см. раздел «Настройка среды HASP HL Net»). Если сообщение об ошибке все равно появляется, причиной могут служить следующие факторы:

- a. Утилита HASP License Manager не загружена.
- b. Если используется протокол TCP/IP, HASP License Manager может находиться в другой подсети.
- c. Если используется протокол IPX, отсутствует поддержка SAP.
- d. Если сообщение об ошибке появляется постоянно, попробуйте применить другой механизм поиска.

## Приложение В

# Глоссарий по HASP HL

<b>Activation counter</b>	<b>Счетчик активаций</b>	Элемент лицензирования, показывающий оставшееся число запусков функции.
<b>AES</b>	<b>Алгоритм AES</b>	AES-алгоритм лежит в основе операций шифрования и дешифрования системы HASP HL.
<b>Aladdin DiagnostiX</b>		Инструмент, применяемый для проверки всей информации, связанной с HASP HL, на компьютере пользователя.
<b>Anti-Debugging</b>	<b>Антиотладочные средства</b>	Средства системы HASP HL, блокирующие возможные атаки, призванные обойти схему защиты.
<b>API Call History</b>	<b>Журнал вызовов API</b>	Журнал, регистрирующий все вызовы HASP HL API, осуществленные HASP HL ToolBox.
<b>API samples</b>	<b>Примеры API</b>	Образцы приложений, использующих HASP HL API. Средство обучения, предназначенное для реализации HASP HL API.
<b>Background checks</b>	<b>Фоновые проверки</b>	Случайные проверки наличия ключа HASP HL, осуществляемые защищенным приложением.

---

<b>Backward compatibility</b>	<b>Обратная совместимость</b>	Возможность применения команд и данных для приложений, защищаемых ключами HASP4. Обратная совместимость HASP HL позволяет считывать и записывать данные, устанавливать часы и пр.
<b>Batch</b>	<b>Серия</b>	Серии применяются для определения функций и пакетов, а также для выполнения команд. Обычно для лицензирования всех ваших продуктов необходима одна серия. Ключи, принадлежащие одной серии, имеют уникальное поведение при шифровании/дешифровании.
<b>Batch Code</b>	<b>Код серии</b>	5-8-значный уникальный код, присваиваемый разработчику. Указывается на наклейке ключа, используется для заказа ключей HASP HL.
<b>Broadcast</b>	<b>Широковещание</b>	Метод обмена данными в сетевой среде, при котором сообщения посылаются каждой рабочей станции.
<b>Broadcast search mechanism</b>	<b>Широковещательный механизм поиска</b>	Механизм поиска, используемый клиентами HASP HL, для поиска HASP License Manager.
<b>C2V file</b>	<b>C2V-файл</b>	Файл, пересылаемый клиентом разработчику. Содержит информацию о статусе ключей HASP HL, находящихся у пользователя.
<b>Concurrent</b>	<b>Одновременное использование</b>	Термин используется по отношению ключей HASP HL или лицензий, показывает одновременность обращения к ключам или лицензиям.
<b>Configuration file</b>	<b>Файл конфигурации</b>	Файл, содержащий информацию о настройках определенной программы. При запуске программы для определения настроек происходит обращение к файлу конфигурации.

---

---

<b>Crack(ing)</b>	<b>Взлом</b>	Намеренный взлом схемы защиты.
<b>Cross platform</b>	<b>Кросс-платформенность</b>	Означает возможность использования функций в среде разных операционных систем.
<b>Daemon</b>	<b>Демон</b>	Фоновый процесс на системах Mac и Linux. Демоны выполняют в операционных системах административные задачи.
<b>Decryption</b>	<b>Дешифрование</b>	Процесс дешифрования данных, которые ранее были зашифрованы.
<b>Default feature</b>	<b>Функция по умолчанию</b>	Всегда доступная функция (при наличии ключа HASP HL). Функция не требует конфигурирования, ее поведение зависит от типа подключенного ключа.
<b>DEMOMA</b>		Серия ключа, используемая для оценки его возможностей. Соответствующий ему Код разработчика находится в папке <i>Vendorcodes</i> .
<b>Demo Vendor Code</b>	<b>Демонстрационный Код разработчика</b>	см. DEMOMA
<b>Encryption</b>	<b>Шифрование</b>	Преобразование данных в секретный код. Шифрование является самым эффективным способом защиты данных. Для дешифрования файла необходим соответствующий крипто-процессор.
<b>Encryption engine</b>	<b>Крипто-процессор</b>	Крипто-процессор ключа HASP HL — основан на применении криптографического алгоритма AES.
<b>Encryption key</b>	<b>Ключ шифрования</b>	Ключ, применяемый для шифрования файла данных, используемого HASP HL Envelope.

---

---

<b>Encryption level</b>	<b>Уровень шифрования</b>	Контролирует компактность вызовов HASP HL при реализации защиты с помощью HASP HL Envelope.
<b>Envelope</b>		см. HASP HL Envelope
<b>Envelope template</b>	<b>Шаблон Envelope</b>	Содержит заданные параметры защиты и прочие данные, связанные с проектом.
<b>Expiration date</b>	<b>Срок действия лицензии</b>	Дата, после которой защищенное приложение не будет работать.
<b>Feature ID</b>	<b>Идентификатор функции</b>	Уникальный идентификатор функции, защищаемой HASP HL. Идентификатор назначается функции с помощью инструмента HASP HL Factory.
<b>Feature Update</b>	<b>Обновление функции</b>	см. update.
<b>File filter</b>	<b>Фильтр файлов</b>	Определяет типы файлов, которые подвергаются шифрованию.
<b>Functionality</b>	<b>Возможности</b>	Действия, которые может осуществлять приложение. Возможности продукта используются маркетологами при идентификации функций продукта.
<b>Handle</b>	<b>Дескриптор</b>	Уникальный идентификатор, служащий для получения доступа к контексту открытой сессии.
<b>HASP</b>		Система защиты и лицензирования программного обеспечения.
<b>HASP HL</b>		Электронный ключ, предназначенный для обеспечения защиты и лицензирования программного обеспечения.

---

---

<b>HASP HL API</b>		Интерфейс, предназначенный для интеграции в программный код приложения вызовов ключа HASP HL.
<b>HASP HL Basic</b>		Стандартный ключ HASP HL. Данная модель не имеет памяти и не может применяться для лицензирования.
<b>HASP HL Demo</b>		Демонстрационный ключ, предназначенный для оценки возможностей защиты и лицензирования с помощью HASP HL.
<b>HASP HL Developer Kit</b>	<b>Демонстрационный Комплект HASP HL</b>	Комплект, содержащий документацию и программное обеспечение системы HASP HL.
<b>HASP HL Device Driver</b>	<b>Драйвер устройства HASP HL</b>	Системный компонент, обеспечивающий обмен данными между защищенным приложением и ключом HASP HL.
<b>HASP HL Envelope</b>		Инструмент, создающий вокруг приложения защитную оболочку. Защищенное таким образом приложение не будет работать, если ключ HASP HL не подключен к компьютеру.
<b>HASP HL Factory</b>		Инструмент, используемый для определения лицензируемых элементов, а также для переноса созданных лицензий на ключи HASP HL.
<b>HASP ID Number</b>	<b>Идентификатор HASP</b>	Уникальный серийный номер ключа HASP HL.
<b>HASP HL Max</b>		Ключ HASP HL с памятью объемом 4КВ. Способен хранить до 112 лицензий.
<b>HASP HL Net</b>		Сетевой ключ HASP HL с памятью объемом 4КВ. Способен хранить до 112 лицензий.

---

<b>HASP HL Pro</b>		Ключ HASP HL с памятью объемом 112 байт. Способен хранить до 16 лицензий.
<b>HASP HL RUS</b>	<b>Система дистанционного перепрограммирования HASP HL</b>	Инструмент, применяемый для удаленного обновления лицензий или изменения содержимого памяти ключа HASP HL.
<b>HASP HL Time</b>		Ключ HASP HL с памятью объемом 4KB и встроенными часами. На ключе может храниться до 8 лицензий, контролирующих срок работы функции, и 104 лицензии, контролирующие количество активаций.
<b>HASP HL ToolBox</b>		Инструмент HASP HL, предназначенный для ознакомления пользователей с HASP HL API, а также для генерирования исходного кода.
<b>HASP License Manager</b>		Действует в качестве сервера и контролирует количество одновременных использований в соответствии с лицензиями, хранящимися на ключе HASP HL Net.
<b>Idle Time</b>	<b>Время бездействия</b>	Интервал времени, по истечении которого станция считается неактивной (применительно к ситуации получения доступа к ключу HASP HL Net).
<b>Key</b>	<b>Ключ</b>	см. HASP HL.
<b>License</b>	<b>Лицензия</b>	Цифровое разрешение, хранящееся на ключе HASP HL. Представляет собой описание прав пользователя по использованию функции.
<b>License Manager</b>		см. HASP License Manager.
<b>Login</b>	<b>Открытие сессии</b>	Операция открытия сессии с ключом HASP HL.

---

<b>Master HASP HL</b>	<b>Мастер-ключ HASP HL</b>	Особый ключ HASP HL, содержащий уникальные коды и идентификаторы, которые были присвоены вам Aladdin. Идентификаторы и коды применяются системой HASP HL при защите и лицензировании программного обеспечения.
<b>Package</b>	<b>Пакет</b>	Термин, описывающий набор функций, управляемых особыми условиями лицензирования. Пакеты определяются с помощью HASP HL Factory.
<b>Program Number</b>	<b>Номер программы</b>	Уникальный идентификатор функции в системе HASP HL.
<b>Real-Time Clock (RTC)</b>	<b>Часы</b>	Часы. Одна из возможностей ключа HASP HL Time.
<b>Remote Update System</b>	<b>Система дистанционного перепрограммирования</b>	Система, применяемая для осуществления безопасного удаленного обновления лицензий, хранящихся на ключах HASP HL.
<b>Reverse Engineering</b>	<b>Рейнжиниринг</b>	Атаки на программное обеспечение, призванные раскрыть алгоритмы защиты приложения путем трассировки скомпилированного приложения к его исходному коду. Система HASP HL обладает средствами противодействия подобным атакам.
<b>Status code</b>	<b>Код статуса</b>	Сообщение о статусе или ошибке, возвращаемое системой HASP HL.
<b>Update</b>	<b>Обновление</b>	Файл, содержащий дополнения к лицензионной информации, хранящейся на ключе HASP HL.
<b>USB</b>		Тип соединения периферийных устройств с компьютером. Используется для подключения ключей HASP HL.
<b>USB Port</b>	<b>Порт USB</b>	Разъем, к которому подключается ключ HASP HL.

---

<b>UTC</b>		Стандартное время, являющееся универсальным для всего мира. Измеряется 24 часами и Грегорианским календарем.
<b>V2C file</b>	<b>V2C-файл</b>	Файл, посылаемый разработчиком конечному пользователю. Содержит данные об обновлении лицензии.
<b>Vendor</b>	<b>Разработчик</b>	Разработчик — это компания, распространяющая программное обеспечение, в которое она вложила интеллектуальные и финансовые ресурсы, и желающая защитить свои инвестиции от неавторизованного и нелегального использования.
<b>Vendor code</b>	<b>Код разработчика</b>	Содержит уникальный идентификатор разработчика, а также всю информацию, необходимую системе HASP HL для поиска и обмена данными с электронным ключом. Код разработчика находится на мастер-ключе HASP HL.
<b>Vendor private key</b>	<b>Закрытый ключ разработчика</b>	Ключ шифрования длиной в 1536 бит, хранящийся на мастер-ключе HASP HL. Используется для добавления цифровой подписи в посылаемые пользователям обновления лицензий или памяти ключей.
<b>Vendor public key</b>	<b>Открытый ключ разработчика</b>	Ключ длиной в 1536 бит, используемый для аутентификации посылаемых пользователю обновлений лицензий или памяти ключей

# Приложение С

# Справочник по HASP HL API

Приложение разделено на три раздела:

- обзор функций, входящих в состав HASP HL API;
- декларации и подробная информация об отдельных функциях HASP HL API;
- описание всех кодов, возвращаемых API.

## **HASP HL ToolBox**

В целях изучения назначения каждого вызова API вы можете использовать HASP HL ToolBox. Данный инструмент позволяет протестировать вызовы функции и требуемые параметры, а также изучить возвращаемые значения. HASP HL ToolBox входит в состав набора программ Vendor Center.

## **Примеры API**

При установке системы HASP HL происходит установка примеров API для различных языков программирования. Эти примеры можно применять при реализации защиты программного кода вашего приложения с помощью HASP HL.

Каждая папка примеров содержит файл заголовков HASP HL. Информацию об имеющихся примерах для определенных языков программирования можно получить на сайте Aladdin или с установочного компакт-диска HASP HL.

## **Обзор функций API**

Перечень функций HASP HL API приводится в таблице С.1.

Таблица С.1 Перечень функций HASP HL API

Функция	Краткое описание
hasp_datetime_to_hasptime()	Изменяет значение даты/времени.
hasp_decrypt()	Дешифрует буфер с помощью алгоритма AES.
hasp_encrypt()	Шифрует буфер с помощью алгоритма AES.
hasp_free()	Освобождает распределенные ресурсы памяти.
hasp_get_rtc()	Считывает время с часов ключа HASP HL Time.
hasp_get_sessioninfo()	Получает информацию о сессии.
hasp_get_size()	Получает информацию о размере файла памяти с ключа HASP HL.
hasp_hasptime_to_datetime()	Изменяет значение времени.
hasp_legacy_decrypt()	Дешифрует буфер, сохраняя обратно совместимое с HASP4 шифрование.
hasp_legacy_encrypt()	Шифрует буфер, сохраняя обратно совместимое с HASP4 шифрование.
hasp_legacy_set_idletime()	Задаёт значение времени простоя для HASP License Manager.
hasp_legacy_set_rtc()	Обратно совместимая функция – устанавливает время на ключе HASP4 Time.
hasp_login()	Открывает сессию, открывает доступ к функции.
hasp_logout()	Закрывает сессию.
hasp_read()	Считывает содержимое памяти ключа HASP HL
hasp_update()	Записывает обновление лицензии HASP HL
hasp_write()	Осуществляет запись в память ключа HASP HL.

### Сведения о HASP HL API

В разделе рассматриваются функции API (для языка программирования С). Информация по функциям разделена на следующие рубрики:

- **Имя:** имя функции для языка С.
- **Описание:** Краткое описание основного назначения функции.
- **Синтаксис:** Декларация функции на языке программирования С.

- **Параметры:** Перечень параметров функции.
- **Возвращаемые ответы:** Перечень всех возможных возвращаемых ответов, связанных с выполнением функции.
- **Применение:** Более подробная информация по применению функции.

## hasp\_datetime\_to\_hasptime()

**Описание**                   Изменяет значение даты/времени и переводит его в формат времени hasp — число секунд, прошедших с 01.01.1970.

### Синтаксис

```
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
)
```

### Параметры

day (день)	ввод значения day (1-31)
month (месяц)	ввод значения month (1-12)
year (год)	ввод значения year (1970+)
hour (час)	ввод значения hour (0-23)
minute (минута)	ввод значения minute (0-59)
second (секунда)	ввод значения секунда (0-59)
time (время)	указатель места значения времени

### Возвращаемые ответы

HASP_STATUS_OK	запрос был успешно выполнен
HASP_INV_TIME	время выходит за рамки поддерживаемых значений

**Применение**               Преобразованное значение показывает количество секунд, прошедших с первого января 1970 г. Эта функция используется в связке с другими функциями API, устанавливающими или получающими значение времени на ключе HASP HL.

**Также смотри**             hasp\_hasptime\_to\_datetime()  
                               hasp\_get\_rtc()  
                               hasp\_legacy\_set\_rtc()

## hasp\_decrypt()

**Описание** Дешифрует буфер с помощью алгоритма AES.

### Синтаксис

```
hasp_status_t HASP_CALLCONV hasp_decrypt(
    hasp_handle_t handle,
    void * buffer,
    hasp_size_t length
)
```

### Параметры

handle (дескриптор)	дескриптор для сессии
buffer (буфер)	указатель буфера, который будет дешифроваться
length (длина)	размер (в байтах) дешифруемого буфера — минимально 6 байт.

### Возвращаемые ответы

HASP_STATUS_OK	запрос был успешно выполнен
HASP_INV_HND	неверный дескриптор
HASP_NOT_IMPL	возможность функции не реализована
HASP_TOO_SHORT	длина дешифруемого буфера слишком коротка
HASP_ENC_NOT_SUPP	ключ не поддерживает тип дешифрования
HASP_CONTAINER_NOT_FOUND	контейнер функции больше не доступен

**Применение** Функция выполняет операцию дешифрования (является обратной по отношению к `hasp_encrypt()`) буфера данных, возвращая его в то состояние, в котором он находился до шифрования. Если операция дешифрования была осуществлена неудачно, дешифруемые данные не утрачиваются.

**Также смотри** `hasp_encrypt()`

## hasp\_encrypt()

**Описание** Шифрует буфер с помощью алгоритма AES.

### Синтаксис

```
hasp_status_t HASP_CALLCONV hasp_encrypt (
    hasp_handle_t handle,
    void * buffer,
    hasp_size_t length
)
```

### Параметры

handle (дескриптор)	дескриптор для сессии
buffer (буфер)	указатель буфера, который будет шифроваться
length (длина)	размер (в байтах) шифруемого буфера – минимально 16 байт.

### Возвращаемые ответы

HASP_STATUS_OK	запрос был успешно выполнен
HASP_INV_HND	неверный дескриптор
HASP_NOT_IMPL	возможность функции не реализована
HASP_TOO_SHORT	длина дешифруемого буфера слишком коротка
HASP_ENC_NOT_SUPP	ключ не поддерживает тип шифрования
HASP_CONTAINER_NOT_FOUND	контейнер функции больше не доступен

**Применение** Осуществляется шифрование данных с помощью крипто-процессора ключа HASP HL. Указанный дескриптор сессии определяет, какой из ключей HASP HL будет осуществлять операцию шифрования буфера данных. Ключ шифрования остается на электронном ключе HASP HL. Если операция шифрования будет выполнена неудачно, шифруемые данные не пострадают. Для дешифрования данных применяется функция `hasp_decrypt`.

**Также смотри** `hasp_decrypt()`

---

## hasp\_free()

**Описание** Освобождает ресурсы памяти, используемые другими функциями API.

**Синтаксис**

```
void HASP_CALLCONV hasp_free(char * info)
```

**Параметры**

info (информация)	указатель на ресурс памяти, который будет освобожден
-------------------	--

**Применение** Применяется для освобождения ресурсов памяти, занятых под хранение данных, полученных в результате вызовов API с помощью `hasp_get_sessioninfo()` и `hasp_update()`. Функция не посылает ответов.

**Смотри также** `hasp_get_sessioninfo()`  
`hasp_update()`

## hasp\_get\_rtc()

**Описание**                    Считывает значение текущего времени с ключа HASP HL Time.

**Синтаксис:**

```
hasp_status_t HASP_CALLCONV hasp_get_rtc(
    hasp_handle_t handle,
    hasp_time_t * time
)
```

**Параметры**

handle (дескриптор)	дескриптор сессии
time (время)	указатель текущего времени

**Возвращаемые  
ответы**

HASP_STATUS_OK	запрос был успешно выполнен
HASP_INV_HND	неверный дескриптор
HASP_NOT_IMPL	возможность функции не реализована
HASP_CONTAINER_NOT_FOUND	контейнер функции больше не доступен
HASP_NO_BATTERY_POWER	у часов закончилось питание
HASP_NO_TIME	часы не доступны

**Применение**                    Назначение функции не связано с лицензированием. Функция применяется преимущественно для получения времени независимо от системных часов. Возвращаемое в ответе значение времени определяется как число секунд, прошедшее с первого января 1970 г. 0:00. Для преобразования значения в дату и время следует применять функцию `hasptime_to_datetime()`.

**Смотри также**                    `hasp_datetime_to_hasptime()`  
    `hasp_hasptime_to_datetime()`

## hasp\_get\_sessioninfo()

**Описание**                   Получает информацию о сессии.

### Синтаксис

```
hasp_status_t HASP_CALLCONV hasp_get_sessioninfo(
    hasp_handle_t handle,
    char * format,
    char ** info
)
```

### Параметры

handle (дескриптор)	дескриптор сессии
format (формат)	XML-определение типа структуры данных. Существует три возможности: <ol style="list-style-type: none"> <li>1. HASP_KEYINFO – формат получения данных о ключе HASP HL.</li> <li>2. HASP_SESSIONINFO – формат получения информации о сессии</li> <li>3. HASP_UPDATEINFO – формат получения информации об обновлении лицензии, которое обычно находится в C2V-файле. Включает информацию по обновлению счетчиков, лицензий, памяти, находящуюся в памяти ключа HASP HL.</li> </ol>
info (информация)	указатель на информацию, получаемую в формате XML-текста

### Возвращаемые ответы

HASP_STATUS_OK	запрос был успешно выполнен
HASP_INV_HND	неверный дескриптор
HASP_INV_FORMAT	неизвестный формат
HASP_INSUF_MEM	недостаточно памяти

**Применение**                   Полученная информация о сессии описывает:

- ключ HASP HL;
- текущую или определенную сессию;
- обновление лицензии.

Данная функция использует память для полученной информации. Для освобождения занятых ресурсов памяти следует применять функцию `hasp_free()`.

**Смотри также**                   `hasp_free()`

---

## hasp\_get\_size()

**Описание**                   Получает сведения о размере файла памяти ключа HASP HL.

**Синтаксис**

```
hasp_status_t HASP_CALLCONV hasp_get_size(  
    hasp_handle_t handle,  
    hasp_fileid_t fileid,  
    hasp_size_t * size  
)
```

**Параметры**

handle (дескриптор)	дескриптор сессии
fileid (идентификатор файла)	идентификатор файла, информация о котором запрашивается
size (размер)	указатель на размер файла

**Возвращаемые  
ответы**

HASP_STATUS_OK	запрос был успешно выполнен
HASP_INV_HND	неверный дескриптор
HASP_INV_FILEID	неизвестный идентификатор файла
HASP_CONTAINER_NOT_FOUND	контейнер функции больше не доступен

**Применение**               Эта функция применяется для определения размера файла памяти HASP HL. Размер файла позволяет определить наибольшее возможное значение смещения. Эта информация необходима при записи или считывании памяти ключа. Первый байт в файле имеет индекс 0.

**Также смотри**             hasp\_read()  
                              hasp\_write()

## hasp\_hasptime\_to\_datetime()

**Описание** Конвертирует значение времени (количество секунд, прошедших с первого января 1970 г.) в дату и время.

### Синтаксис

```
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
)
```

### Параметры

time (время)	указатель места значения времени
day (день)	указатель значения day
month (месяц)	указатель значения month
year (год)	указатель значения year
hour (час)	указатель значения hour
minute (минута)	указатель значения minute
second (секунда)	указатель значения second

### Возвращаемые ответы

HASP_STATUS_OK	запрос был успешно выполнен
HASP_INV_TIME	время выходит за рамки поддерживаемых значений

**Применение** Все значения времени основываются на времени UTC. Преобразованное значение показывает количество секунд, прошедших с первого января 1970 г. Эта функция используется в связке с другими функциями API, устанавливающими или получающими значение времени с ключа HASP HL.

**Также смотри** `hasp_datetime_to_hasptime()`

## hasp\_legacy\_decrypt()

**Описание** Дешифрует буфер, сохраняя обратную совместимость с HASP4.

### Синтаксис

```
hasp_status_t HASP_CALLCONV hasp_legacy_decrypt (
    hasp_handle_t handle,
    void * buffer,
    hasp_size_t length )
)
```

### Параметры

handle (дескриптор)	дескриптор для сессии
buffer (буфер)	указатель буфера, который будет дешифроваться
length (длина)	размер (в байтах) дешифруемого буфера — минимально 8 байт.

### Возвращаемые ответы

HASP_STATUS_OK	запрос был успешно выполнен
HASP_INV_HND	неверный дескриптор
HASP_TOO_SHORT	длина дешифруемого буфера слишком коротка
HASP_ENC_NOT_SUPP	ключ не поддерживает тип дешифрования
HASP_CONTAINER_NOT_FOUND	контейнер функции больше не доступен

**Применение** Выполнение функции зависит от дескриптора, сгенерированного вызовом `hasp_login()`. Идентификатор функции должен быть задекларирован как «PROGRAM NUMBER FEATURE». Если операция дешифрования не заканчивается успешно, данные, указываемые буфером, остаются неопределенными.

**Смотри также** `hasp_legacy_encrypt()`

## hasp\_legacy\_encrypt()

**Описание**                      Шифрует буфер с сохранением обратной совместимости с HASP4.

### Синтаксис

```
hasp_status_t HASP_CALLCONV hasp_legacy_encrypt(
    hasp_handle_t handle,
    void * buffer,
    hasp_size_t length
)
```

### Параметры

handle (дескриптор)	дескриптор для сессии
buffer (буфер)	указатель буфера, который будет шифроваться
length (длина)	размер (в байтах) шифруемого буфера – минимально 8 байт.

### Возвращаемые ответы

HASP_STATUS_OK	запрос был успешно выполнен
HASP_INV_HND	неверный дескриптор
HASP_NOT_IMPL	возможность функции не реализована
HASP_TOO_SHORT	длина дешифруемого буфера слишком коротка
HASP_ENC_NOT_SUPP	ключ не поддерживает тип шифрования
HASP_CONTAINER_NOT_FOUND	контейнер функции больше не доступен

### Применение

Выполнение функции зависит от дескриптора, сгенерированного вызовом `hasp_login()`. Идентификатор функции должен быть задекларирован как «PROGRAM NUMBER FEATURE». Если операция шифрования не заканчивается успешно, данные, указываемые буфером, остаются неопределенными.

### Смотри также

`hasp_legacy_decrypt()`

---

## hasp\_legacy\_set\_idletime()

**Описание**                    Задает значение времени простоя для HASP License Manager.

### Синтаксис

```
hasp_status_t HASP_CALLCONV hasp_legacy_set_idletime (  
    hasp_handle_t handle,  
    hasp_u16_t idle_time  
)
```

### Параметры

handle (дескриптор)	дескриптор для сессии
idle_time (время простоя)	значение idle time в минутах

### Возвращаемые ответы

HASP_STATUS_OK	запрос был успешно выполнен
HASP_INV_HND	неверный дескриптор
HASP_REQ_NOT_SUPP	попытка указать время простоя для локального ключа
HASP_CONTAINER_NOT_FOUND	контейнер функции больше не доступен

**Применение**                Выполнение функции зависит от дескриптора, сгенерированного вызовом hasp\_login(). Идентификатор функции должен быть задекларирован как «PROGRAM NUMBER FEATURE». Функция используется только при получении доступа к ключу HASP HL через сеть.

**Смотри также**             hasp\_login()

## hasp\_legacy\_set\_rtc()

**Описание**                      Функция позволяет установить значение часов ключа с сохранением обратной совместимости с HASP4.

### Синтаксис

```
hasp_status_t HASP_CALLCONV hasp_legacy_set_rtc(
    hasp_handle_t handle,
    hasp_time_t new_time)
)
```

### Параметры

handle (дескриптор)	дескриптор для сессии
new time (новое время)	значение времени для установки часов

### Возвращаемые ответы

HASP_STATUS_OK	запрос был успешно выполнен
HASP_INV_HND	неверный дескриптор
HASP_CONTAINER_NOT_FOUND	контейнер функции больше не доступен
HASP_NO_TIME	часы недоступны или ключ подключен не к локальному компьютеру

**Применение**                      Выполнение функции зависит от дескриптора, сгенерированного вызовом `hasp_login()`. Идентификатор функции должен быть задекларирован как «PROGRAM NUMBER FEATURE». Для преобразования значений времени следует использовать соответствующие функции.

**Смотри также**                      `hasp_get_rtc()`  
    `hasp_hasptime_to_datetime()`  
    `hasp_datetime_to_hasptime()`

## hasp\_login()

**Описание** Подключается к функции и таким образом открывает сессию.

### Синтаксис

```
hasp_status_t HASP_CALLCONV hasp_login(
    hasp_feature_t feature_id,
    hasp_vendor_code_t vendor_code,
    hasp_handle_t * handle
)
```

### Параметры

feature id (идентификатор функции)	<p>Уникальный идентификатор определенной функции, хранящийся на ключе HASP HL. Функция задает параметры открытия сессии следующим образом:</p> <ol style="list-style-type: none"> <li>1. «PROGRAM NUMBER FEATURE» указывает, что происходит получение доступа к определенному программному номеру.</li> <li>2. Уникальное значение программного номера.</li> <li>3. Особые характеристики сессии: <ul style="list-style-type: none"> <li>• лицензии только с локального ключа;</li> <li>• лицензии только с ключа, подключенного к другой сетевой станции;</li> <li>• значение счетчика уменьшается по количеству процессов;</li> <li>• отключение проверки терминального сервера (позволяет запускать приложения на станциях терминального сервера);</li> <li>• разрешить использование ключей HASP3/HASP4.</li> </ul> </li> </ol>
vendor code (код разработчика)	указатель кода разработчика
handle (дескриптор)	указатель дескриптора сессии

**Возвращаемые  
ответы**

HASP_CONTAINER_NOT_FOUND	ключ не найден
HASP_STATUS_OK	запрос был успешно выполнен
HASP_FEATURE_NOT_FOUND	запрашиваемая функция не найдена
HASP_FEATURE_TYPE_NOT_IMPL	запрашиваемый тип функции недоступен
HASP_INV_PROGNUM_OPT	запрошена неизвестная опция «PROGRAM NUMBER FEATURE»
HASP_TMOF	слишком много открытых дескрипторов
HASP_INSUF_MEM	недостаточно памяти
HASP_INV_VCODE	неверный код разработчика
HASP_NO_DRIVER	драйвер не установлен
HASP_OLD_DRIVER	установлен старый драйвер
HASP_TS_DETECTED	программа работает удаленно на терминальном сервере

**Применение**

Данная функция открывает сессию с ключом HASP HL, на котором находится лицензия с запрашиваемым номером программы.

Идентификатор функции соответствует «PROGRAM NUMBER FEATURE», находящемуся на ключе HASP HL.

При использовании функции по умолчанию (или программного номера «0») API осуществляет поиск исключительно самого ключа HASP HL, а не лицензионной информации.

Соответствующие коды разработчика находятся на вашем компьютере в папке *Vendorcodes*. Вызов функции без верного кода разработчика невозможен. Одновременно можно открывать до 128 сессий.

**Смотри также**

hasp\_logout()

## hasp\_logout()

**Описание**                      Завершает открытую сессию.

**Синтаксис**

```
hasp_status_t HASP_CALLCONV hasp_logout (
    hasp_handle_t handle
)
```

**Параметры**

handle (дескриптор)	дескриптор для сессии
---------------------	-----------------------

**Возвращаемые  
ответы**

HASP_STATUS_OK	запрос был успешно выполнен
HASP_INV_HND	неверный дескриптор

**Применение**                      Данная функция используется для разрыва связи с объектом API. По окончании сессии вся используемая в ее процессе память освобождается. Связь с HASP License Manager прекращается, если последней сессией API была logout.

**Смотри также**                      hasp\_login()

## hasp\_read()

**Описание**                    Считывает память ключа HASP HL.

### Синтаксис

```
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
)
```

### Параметры

handle (дескриптор)	дескриптор для сессии
fileid (идентификатор файла)	идентификатор запрашиваемого файла
offset (смещение)	байтовое смещение файла
length (длина)	число байтов файла
buffer (буфер)	указатель получаемого файла

### Возвращаемые ответы

HASP_STATUS_OK	запрос был успешно выполнен
HASP_INV_HND	неверный дескриптор
HASP_NOT_IMPL	запрашиваемый тип функции недоступен
HASP_INV_FILEID	неизвестный идентификатор файла
HASP_MEM_RANGE	недостаточно памяти
HASP_CONTAINER_NOT_FOUND	контейнер функции больше не доступен

**Описание**                    Доступны следующие опции идентификатора файла:  
                                   HASP\_FILEID\_LICENSE  
                                   HASP\_FILEID\_MAIN.

Доступ зависит от дескриптора, сгенерированного `hasp_login()`. Идентификатор функции должен декларироваться как «PROGRAM NUMBER FEATURE», что запрещает доступ к файлу памяти HASP4.

Для определения размера считываемого из памяти файла следует применять функцию `hasp_get_size()`.

**Смотри также**

`hasp_get_size()`

`hasp_write()`

## hasp\_update()

**Описание**                                Записывает обновление лицензии в память ключа HASP HL.

### Синтаксис

```
hasp_status_t HASP_CALLCONV hasp_update (
    char * update_data,
    char ** ack_data
)
```

### Параметры

update_data (данные обновления)	указатель на полные данные обновления
ack_data (подтверждающие данные)	указатель на буфер для получения подтверждающих данных

### Возвращаемые ответы

HASP_INV_UPDATE_DATA	требуемые метки XML не найдены ИЛИ данные в двоичной форме отсутствуют или повреждены
HASP_INV_UPDATE_OBJ	данные в двоичном коде не содержат обновления
HASP_NO_ACK_SPACE	обновление запрашивает подтверждающие данные, однако значением параметра ack_data является NULL
HASP_KEYID_NOT_FOUND	обновляемая лицензия на ключе HASP HL не найдена
HASP_INV_UPDATE_NOTSUPP	обновление не поддерживается ключом HASP HL
HASP_UNKNOWN_ALG	в V2C-файле используется неизвестный алгоритм
HASP_INV_UPDATE_CNTR	счетчик обновления установлен некорректно
HASP_INV_SIG	не пройдена проверка подписи

### Применение

Функция записывает информацию об обновлении.

Код обновления содержит все необходимые данные для собственного переноса на ключ HASP HL:

1. место записи (на какой ключ HASP HL);

2. данные, необходимые для получения доступа (Код разработчика);

3. данные обновления.

Функция возвращает код подтверждения, шифруемый/подписываемый обновлением. Код является свидетельством того, что обновление было применено к лицензии. Память под подтверждение распределяется API и должна освобождаться с помощью `hasp_free()`.

Эта функция является расширением HASP HL API и применяется HASP HL RUS при обновлении лицензий.

**Смотри также**

`hasp_free()`

## hasp\_write()

**Описание**                    Осуществляет запись в память ключа HASP HL.

### Синтаксис

```
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
)
```

### Параметры

handle (дескриптор)	дескриптор для сессии
fileid (идентификатор файла)	идентификатор записываемого файла
offset (смещение)	байтовое смещение файла
length (длина)	число байтов в файле
buffer (буфер)	указатель получаемых данных

### Возвращаемые ответы

HASP_STATUS_OK	запрос был успешно выполнен
HASP_INV_HND	неверный дескриптор
HASP_NOT_IMPL	запрашиваемый тип функции недоступен
HASP_INV_FILEID	неизвестный идентификатор файла
HASP_MEM_RANGE	недостаточно памяти
HASP_ACCESS_DENIED	доступ к функции закрыт
HASP_CONTAINER_NOT_FOUND	контейнер функции больше не доступен

**Применение**                    Для записи данных в настоящее время доступно поле HASP\_FILEID\_MAIN.

Доступ зависит от дескриптора, сгенерированного `hasp_login()`. Идентификатор функции должен декларироваться как «PROGRAM NUMBER FEATURE», что запрещает доступ к файлу памяти HASP4.

**Смотри также**

`hasp_get_size()`  
`hasp_read()`

---

## Коды статуса API

Ниже в алфавитном порядке приводится перечень и описание возможных кодов, связанных с работой функций HASP HL API.

№	Код статуса	Описание
0	HASP_STATUS_OK	Запрос был успешно выполнен
1	HASP_MEM_RANGE	Ваш запрос выходит за рамки размеров памяти HASP HL.
2	HASP_INV_PROGNUM_OPT	Вы запросили неизвестный «PROGRAM NUMBER FEATURE».
3	HASP_INSUF_MEM	Вашей системе недостаточно памяти.
4	HASP_TIMEOF	Слишком много открытых дескрипторов.
5	HASP_ACCESS_DENIED	В доступе к функции отказано.
6	HASP_INCOMPAT_FEATURE	Дешифрование не может быть применено к данной функции.
7	HASP_CONTAINER_NOT_FOUND	Ключ HASP HL недоступен.
8	HASP_TOO_SHORT	Размер дешифрованных данных недостаточен для выполнения вызова функции.
9	HASP_INV_HND	Для функции был неверно указан дескриптор.
10	HASP_INV_FILEID	Указанный идентификатор не распознается API.
11	HASP_OLD_DRIVER	Установленный устаревший драйвер не позволяет выполнить функцию.
12	HASP_NO_TIME	Часы ключа не доступны.
13	HASP_SYS_ERROR	Ошибка вызова системы
14	HASP_NO_DRIVER	Необходимый драйвер не установлен.
15	HASP_INV_FORMAT	Нераспознаваемый формат файла обновления.
16	HASP_REQ_NOT_SUPP	Функция не может быть выполнена в данной сессии.
17	HASP_INV_UPDATE_OBJ	Данные в двоичном коде не содержат обновления.
18	HASP_KEYID_NOT_FOUND	Обновляемая лицензия на ключе HASP HL не найдена.

19	HASP_INV_UPDATE_DATA	Необходимые XML-маркеры не найдены. Часть данных в двоичном коде утеряна или повреждена.
20	HASP_INV_UPDATE_NOTSUPP	Запрос на обновление не поддерживается ключом HASP HL.
21	HASP_INV_UPDATE_CNTR	Счетчик обновления установлен неверно.
22	HASP_INV_VCODE	Вы применили неверный код разработчика.
23	HASP_ENC_NOT_SUPP	Используемый ключ не поддерживает данный тип шифрования.
24	HASP_INV_TIME	Полученное значение времени выходит за поддерживаемые границы.
25	HASP_NO_BATTERY_POWER	Разряжена батарея часов электронного ключа.
26	HASP_NO_ACK_SPACE	Обновление запрашивает подтверждающие данные, однако значением параметра ack_data является NULL
27	HASP_TS_DETECTED	Программа запущена на терминальном сервере.
28	HASP_FEATURE_TYPE_NOT_IMPL	Не реализован тип запрошенной функции.
29	HASP_UNKNOWN_ALG	В V2C-файле используется неизвестный алгоритм.
30	HASP_INV_SIG	Не пройдена операция проверки подписи.
31	HASP_FEATURE_NOT_FOUND	Запрошенная функция недоступна.
500	HASP_INVALID_OBJECT	Объект инициализирован некорректно.
501	HASP_INVALID_PARAMETER	Ошибка в параметре.
502	HASP_ALREADY_LOGGED_IN	Повторное открытие сессии с тем же объектом.
503	HASP_ALREADY_LOGGED_OUT	Повторное закрытие сессии с тем же объектом.
525	HASP_OPERATION_FAILED	Ошибка, связанная с некорректной работой системы или платформы.
698	HASP_NOT_IMPL	Запрашиваемый тип функции не реализован.

# Спецификации ключей HASP HL

Таблица D.1 Технические спецификации ключей HASP HL

Характеристика	«Большие» HASP HL	«Мини» HASP HL
Типы моделей	HASP HL Time, HASP HL Net, HASP HL Max, HASP HL Pro, HASP HL Basic, Master HASP HL	HASP HL Net, HASP HL Max, HASP HL Pro, HASP HL Basic
Длина	52 мм	38 мм
Высота	8 мм	8 мм
Ширина	16 мм	16 мм
Вес	5,4 гр. 6,1 гр. (HASP HL Time)	4,4 гр.
Тип разъема	USB тип А	USB тип А
Используемые линии	Питание, земля, 2 для данных	Питание, земля, 2 для данных
Материал пластикового корпуса	Lexan 500R	Lexan 500R
Рабочая температура	-25°C-70°C -13°F-158°F	-25°C-70°C -13°F-158°F

Срок хранения данных в памяти	не менее 10 лет	не менее 10 лет
Число циклов перезаписи памяти	1 000 000	1 000 000
цвет LED-индикатора	красный	красный
Влажность	0-100% без конденсации	0-100% без конденсации
Энергопотребление	Рабочее: 50 мА Режим ожидания: <0,5 мА	Рабочее: 50 мА Режим ожидания: <0,5 мА

**Таблица D.2 Технические характеристики моделей HASP HL**

Характеристика	HASP HL Basic	HASP HL Pro	HASP HL Max	HASP HL Net	HASP HL Time
Размер памяти, байт	-	112	4032	4032	4032
Срок работы батареи	-	-	-	-	4 года