圣天诺LDK快速入门

# 概述

圣天诺LDK是泰雷兹提供的一套功能强大、简单易用的软件保护与许可授权解决方案，支持多种硬件平台与操作系统，包括Windows、苹果Mac以及X86与ARM架构下的Linux与Android。LDK同时支持硬许可与软许可。

LDK提供两种保护功能：

* **保护**：即通过LDK提供的**外壳**（Envelope）技术，保护软件的二进制代码不被反编译、逆向工程，从而保护软件的知识产权；
* **许可**：即通过在软件源代码里嵌入LDK **API**，实现多种类型的许可（License）功能，如软件某个功能只允许使用90天，等等。

下面首先介绍一下什么是许可及许可的类型。

# 许可（License）

## 2.1 基本概念

许可是软件某个特征（Feature）或功能的使用条件，符合条件，该特征才可以被使用。特征通过Feature ID唯一标示，特征与许可是一一对应的关系。对于用外壳保护的程序，其许可对应的特征是整个程序（包括所有功能）。API可以实现基于特征的许可授权。

下面是一个用于多功能打印机软件许可的例子。

|  |  |  |
| --- | --- | --- |
| **特征ID** | **特征名称** | **使用条件** |
| 11 | Print | 永久 |
| 12 | Copy | 365天 |
| 13 | Fax | 1000次执行 |

程序保护（无论是通过外壳还是API）与许可定义是分开进行的，许可发生变化（如Copy功能从365天改为永久），程序无需重新被保护。这就是 “一次保护，多次分发”，LDK的基本设计哲学。

## 2.2 许可类型

按许可的存储介质分类，包括硬许可与软许可两类：

* **硬许可（HL，又称为硬锁或加密狗）**

通常是USB类型的硬件，许可信息保存在里面。

* **软许可（SL，又称为软锁）**

通常是一个加密文件，许可信息同时还保存在硬盘的一个隐含分区。

根据是否与机器硬件绑定，软锁可分为两种类型：

* 绑定机器的软锁：软锁与机器的硬件指纹信息（如CPU、硬盘、MAC地址等）绑定，如拷贝到其他机器上，软锁将失效并报克隆的错误。安装软锁的过程称为**激活**，有离线与在线两种方式：
* **离线方式：**用户目标机器无法连接互联网的情况，即无法连接到后台EMS，过程如下：1）用RUS工具收集目标机器机器硬件指纹信息（. C2V文件）；2）EMS使用该C2V文件生成许可（.V2C文件）3）在目标机器安装v2c文件。
* **在线方式：**需要目标机器可以连接互联网，过程比较简单，只需通过产品验证码（Product Key）登录EMS客户门户网站（Customer portal），点击激活即可，详细情况参见第三节EMS部分。
* 不绑定机器的软锁：与机器硬件信息无关，可以复制到其他机器上使用，通常用于试用授权。这种软锁无需生成指纹，直接安装V2C即可。

根据功能权限划分，软锁又包括两种类型：

* 管理员软锁（AdminMode SL）：目标机器需安装运行环境RTE（或称为驱动），可以通过开发商套件中的ACC工具（http://localhost:1947）进行管理；功能比较丰富，如作为网络锁控制客户端程序的并发数，支持软锁从一台机器上迁移到另外一台等。
* 用户软锁（UserMode SL）：目标机器无需安装RTE，无法通过ACC管理，功能相对简单。

## 2.3 许可模型

按使用条件分，许可包括下面几类（又称为许可模型）：

* 永久许可：对应的程序特征永远有效
* 时间段许可：在某段时间内有效，如30天。
* 绝对时间许可：在某个特定日期前有效，如到2019.1.1。
* 执行次数许可：程序特征只能执行指定次数，如100次。
* 网络许可（或并发许可）：许可安装在一个服务器（称为License Manager），客户端程序通过网络（通常为局域网）访问该许可，许可可以控制好客户端程序的并发数。

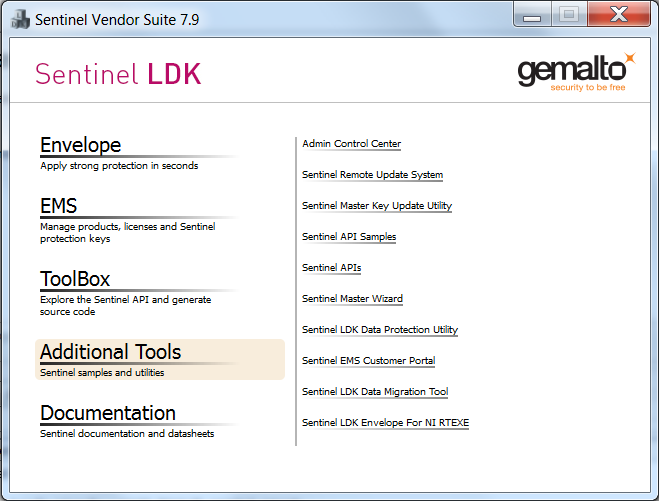
# 2. LDK开发工具

圣天诺LDK安装光盘（或压缩安装包）包括外壳工具（Envelope）、License API库、后台授权管理系统EMS以及其他一些开发工具。

## 2.1 Windows下开发工具

您（软件开发商或智能设备提供商）至少需要一台Windows机器，用于保护Windows程序与Android应用。开发包安装非常简单，在DVD（或iso解压缩）下的Windows目录，以管理员权限运行setup，一路选择缺省（除非打算在另外一台Windows机器上安装EMS）。

安装后打开开发商套件（Vendor Suite），如下：



* Envelope：外壳工具
* EMS：基于Web的后台授权管理系统（详见第3节）
* ToolBox：API使用辅助工具
* Additional Tools：附加工具
* Documentation：相关文档

外壳Envelope、授权管理系统EMS以及API工具ToolBox下文有介绍。这里简要介绍一下附加工具（Additional Tools）。附加工具提供了几个开发商和用户使用的工具，常用的工具包括：

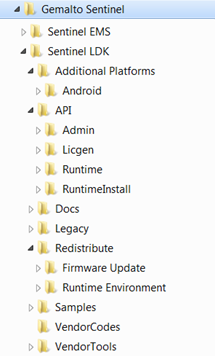
* Admin Control Center （ACC，即管理控制中心）：用户端工具，基于Web界面，用于查看、管理用户机器上所有的许可；
* Sentinel Remote Update System （即RUS，远程升级系统）：用户端工具，用于收集用户的机器指纹、已安装的许可信息以及安装、升级许可等；
* Sentinel Master Key Update Utility （主锁升级工具）：开发商工具，用于更新主锁（蓝色的USB）；
* Sentinel API Samples：开发商工具，打开APIs示例代码目录；
* Sentinel APIs：开发商工具，打开APIs即SDK库文件目录；
* Sentinel Master Wizard（主锁导入向导）：开发商工具，导入Vendor Code与SDK库（包括各类API）；
* Sentinel Data Protection Utility （数据保护工具）：开发商工具，用于加密数据文件（外壳是用户加密程序），注意Linux X86/ARM系统下的数据文件加密也在Windows机器上进行。
* Sentinel EMS Customer Portal（EMS客户门户）：用户端工具，EMS提供给用户的Web门户，用于查看、激活许可等。

注：我们给每个开发商提供一个与特定秘钥相关的开发商代码（Vendor Code）以及用该秘钥加密的SDK库。安装完成后，在机器上插入主锁，运行主锁导入向导（Sentinel Master Wizard），即可下载您的Vendor Code与SDK库，缺省放在用户目录MyDocuments\ Gemalto\Sentinel LDK 7.9下或C:\Program Files (x86)\Gemalto Sentinel\Sentinel LDK。

对于开发人员，因为不需要为客户生成授权，可以使用开发锁（Developer Key）对程序加壳。

安装后的目录结构（缺省在C:\Program Files (x86)\Gemalto Sentinel）：

* Android：Android开发工具包，包括API、文档、示例程序等。
* API：Windows下API（SDK库），包括许可管理（Admin）、许可生成Licgen、许可操作相关的运行时等（Runtime）、运行时环境安装等API。
* Docs：文档
* Redistribute：给用户分发的软件工具，包括硬锁固件升级、运行时环境（Runtime Environment，即RTE）等。
* Samples：API示例程序
* VendorCodes：用于测试的开发商代码，保存于.hvc文件。
* VendorTools：开发商工具，如外壳等。

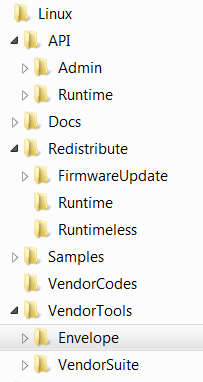


## 2.2 Android下开发工具

Android开发在Windows平台下进行。在Windows安装目录，有个Additional Platforms\Android子目录，包含API库、文档、示例代码等。Android外壳工具与Windows相同。

## 2.3 Linux下开发工具

Linux下没有需要安装的软件，只需要将在DVD（或iso解压缩）下Linux全部子目录、文件拷贝到Linux机器上即可。注意对于目标机器是Linux ARM的情况，开发工作如加壳、API编程也要在X86机器上进行。Linux目录结构如下：

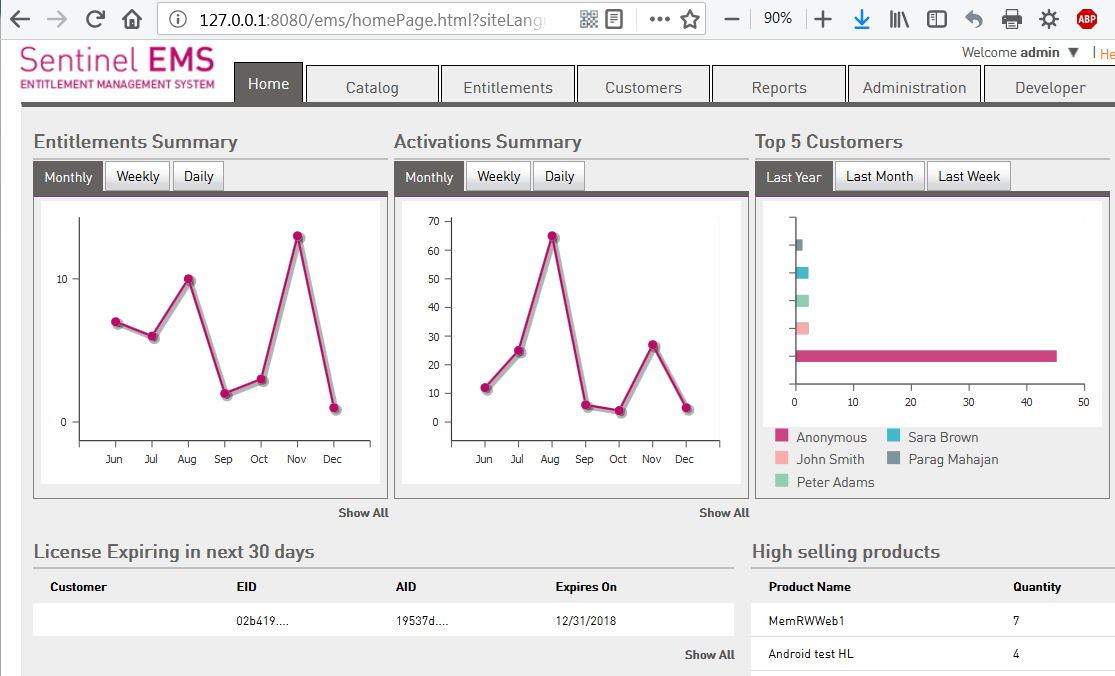


* API：API库，包括许可管理库（Admin）与运行时库（Runtime）
* Docs：文档
* Resdistribute：给用户分发的软件工具，包括硬锁固件升级、运行时环境（Runtime Environment，即RTE）等。
* Samples：API示例程序
* VendorCodes：开发商代码
* VendorTools：开发商工具，包括外壳、主锁导入向导等。

# 授权管理系统（EMS）

## EMS简介

EMS部署在您（开发商）机房或数据中心的后台授权管理系统。EMS按登录用户的角色，提供不同的管理功能视图。EMS是基于Web的，用户可以通过浏览器访问。



EMS最基本的功能是创建授权，生成许可，因为许可关联到特征（Feature），因此还提供特征等产品目录（Catalog）管理功能。在EMS上可以创建特征、将一个或多个特征打包成产品（Product）以及创建授权（Entitlement）等操作。具体步骤详见下一节介绍。

## 3.2 许可生成

许可生成（对于硬锁，又称为烧锁）有三种方式：

* EMS上生成
* 用基于EMS Web Service APIs的程序自动生成
* 用基于License API (licgen) 的程序自动生成

**下面介绍通过EMS如何生成、安装软许可。**

1. 获取机器指纹（适用于离线激活的情况，在线激活无需这一步）。

**Windows系统**

在目标机器上，使用RUS工具（rus.exe，默认在C:\Program Files (x86)\Gemalto Sentinel\Sentinel LDK\VendorTools\VendorSuite），选择**收集状态信息**页面，点击**收集信息**按钮，生成一个.C2V文件。



***注意：***

* *如本机上没有安装任何软锁，选择****安装新保护锁****，否则点击****更新现有保护锁****（需选择一个现有的软锁）。*
* *开发商套件自带的RUS对应的开发商代码是DEMOMA，只用于测试。如果您已经购买并导入了主锁，有了正式的开发商号，请使用EMS生成一个正式号的RUS工具（在Developer页面下的RUS Branding子页面）。*

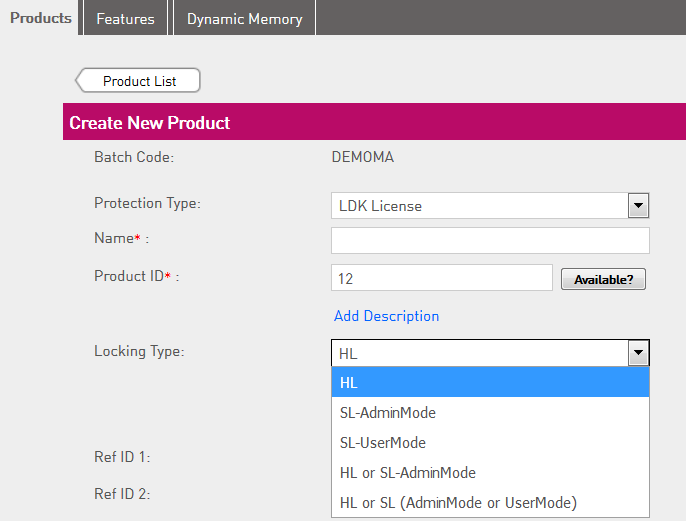
**Linux系统**

进入目录.\Samples\Runtime\C\gcc，你需要根据自己CPU的类型，选择其中一个子目录：如ARM64位系统进入arm64，32位软浮点系统进入armel，32位硬浮点则进入armhf。然后进入static（shared也可以），使用命令hasp\_update进行取指纹或安装软锁操作，命令格式如下：

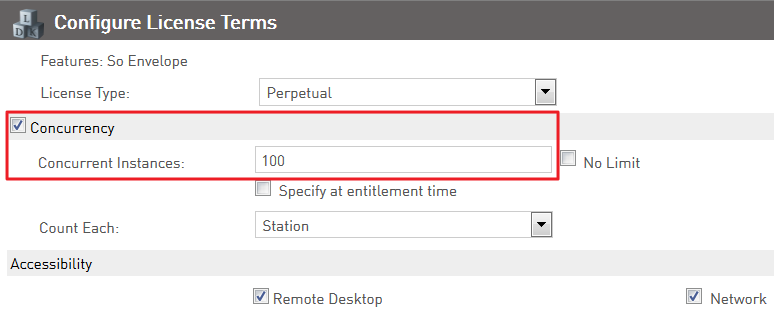
* hasp\_update i xxx. C2V，用于在已经装有一个软锁的系统下取指纹，并存为xxx. C2V；
* hasp\_update f xxx. C2V，用于在没有安装任何软锁的系统下取指纹，并存为xxx. C2V；
* Hasp\_update u xxx.V2C，用于安装（也称为激活）一个软锁，xxx.V2C是在EMS上离线激活软锁时生产的（详见7至13步）。

*注意：上述hasp\_update命令只能用在开发商代码为DEMOMA的测试程序。如果是使用你们自己公的开发商代码，则需要从你们公司的API库重新编译.\Samples\Runtime\C下的hasp\_update.c文件。建议在目前的测试阶段使用DEMOMA。*

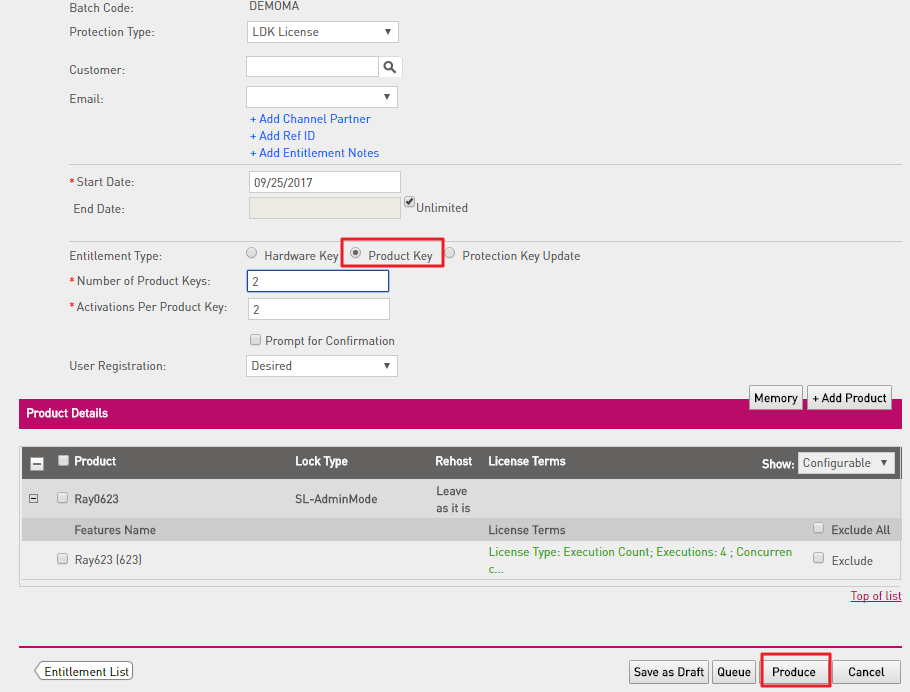
1. 在开发商套件中打开EMS（如EMS安装在本机，在浏览器打开 <http://127.0.0.1:8080/ems/login.html>）。点击Catalog页面，创建特征（Feature），注意要与程序加壳时使用的Feature ID一致（如100）。EMS默认使用测试开发号，即DEMOMA，如果使用正式开发号创建授权，请在batch code里选择对应的开发号。
2. 在EMS的Catalog页面，创建产品（Product），加入上一步创建的Feature。注意锁定类型（Locking Type）要包含SL-AdminMode或SL-UserMode。



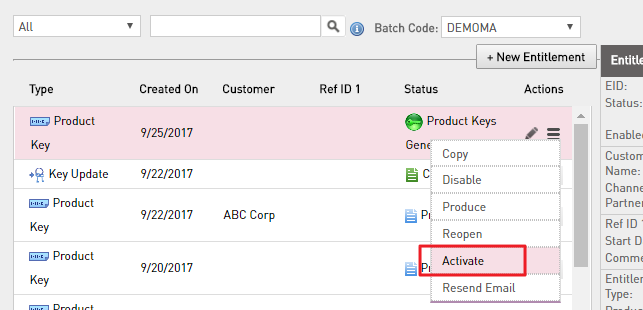
1. 在Feature后面，点击Configure，配置License属性。如下面配置该Feature为并发License（又称为网络锁，最多允许100个终端访问）。



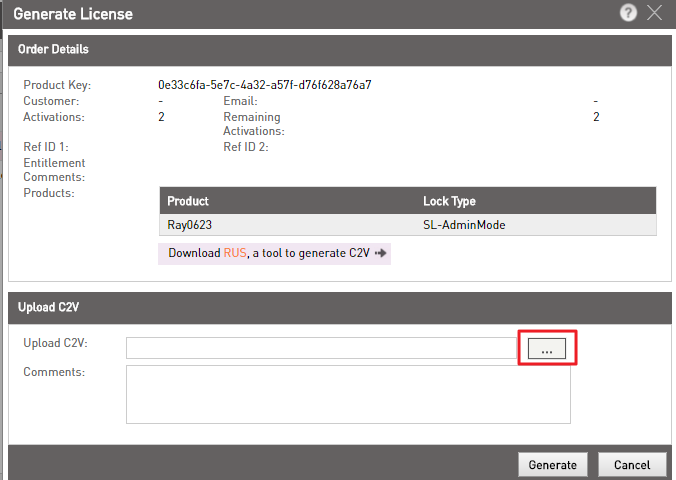
1. 创建授权（Entitlement），授权类型指定为Product Key（即软许可），并加入上一步创建的的Product。



1. 激活许可（下面页面是开发商帮助客户激活许可的情况，客户也可以自助激活，见3.3节）。



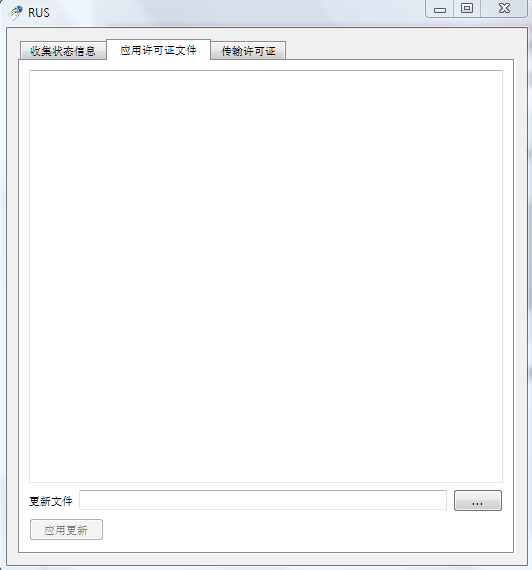
1. 上载第4步产生的指纹文件（myfp.C2V）点击生成（Generate），生成一个License文件（.V2C），如105018327607189619.V2C。



1. 在目标机器上安装V2C文件。

**Windows系统**

在目标机器上打开RUS工具（见第1步），选择**应用许可证文**件页面，点击…，选中上一步生成的V2C文件，然后点击**应用更新**按钮。

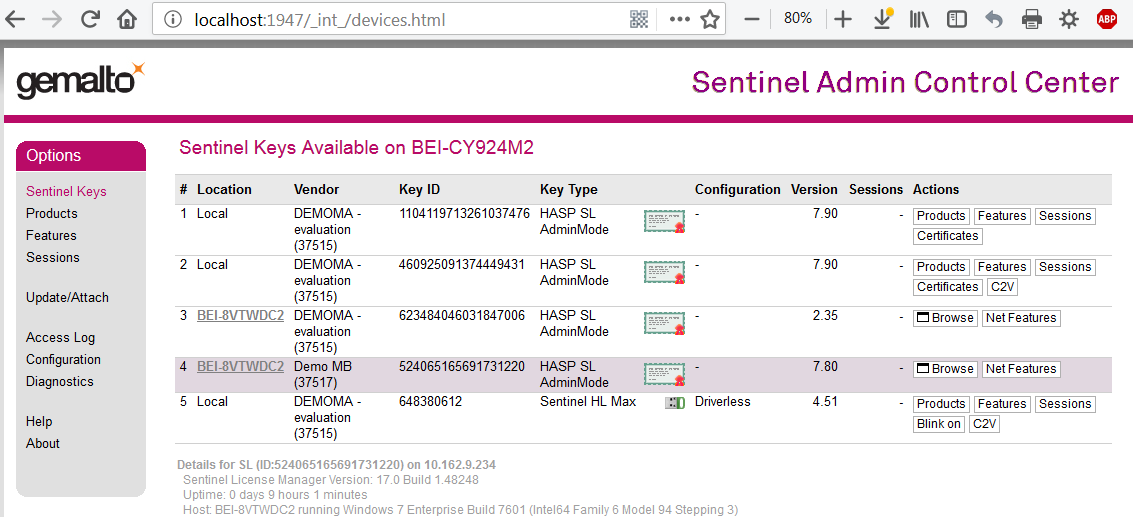


**Linux系统**

进入Gemalto/Samples/Runtime/C/gcc/x86\_64/static目录，运行hasp\_update（Linux下RUS工具，只有命令行方式。假如目标平台不是x86\_64，则在gcc下选择其他目录，或用自己的工具链编译hasp\_update.c代码）：

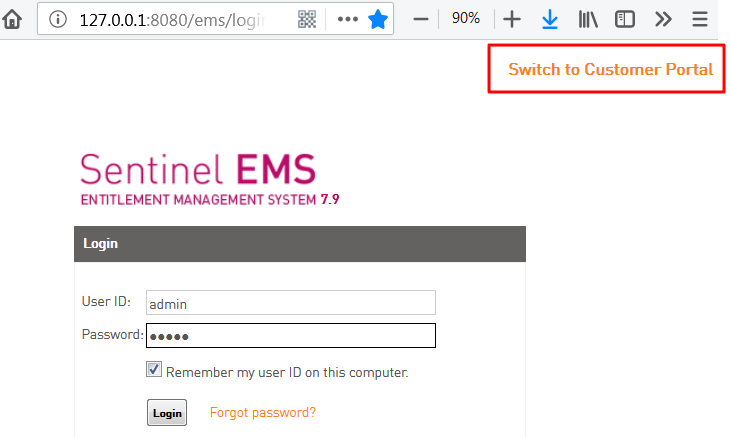
./hasp\_update u 105018327607189619.V2C

1. 运行保护的程序。
2. 如果Linux主机有图形用户界面，在开发商套件中打开ACC工具（或在浏览器打开<http://localhost:1947>）查看并管理（如安装、迁移等）这台机器的许可，如下图：



## 客户门户

EMS提供了一个客户门户网站，允许用户自助激活软许可。用户在浏览器上课通过产品密钥（ProductKey）­登录客户门户，进行软许可激活操作。



激活有在线与离线两种方式（如下图）。**在线激活**是目标机器上安装软许可，适用于目标机器可通过网络连接到EMS的情况。**离线激活**是在一台联网机器，输入目标机器硬件指纹（C2V文件），生成V2C文件，然后拷贝到目标机器上进行安装。

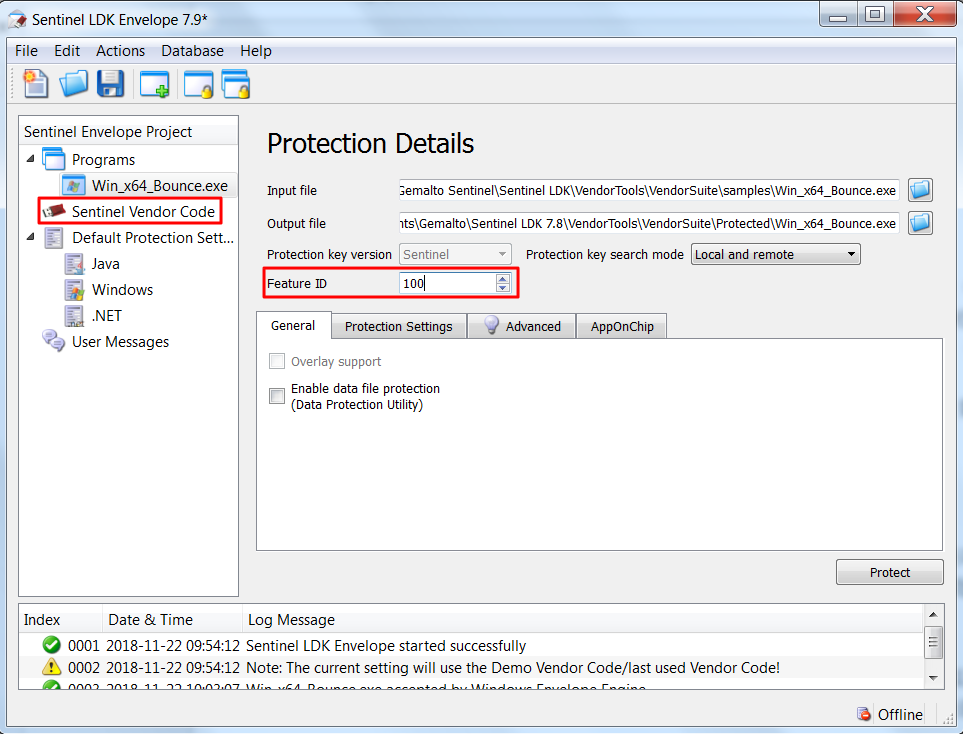


# 外壳（Envelope）

对整个执行程序（如.exe、.lib、.dll以及.a、.so、.apk）进行加密，并对代码进行混淆（可选）。通过在程序外层添加一层“外壳代码”，检测机器上是否有跟踪调试与内存转储等程序，此外还可以检查许可（license），如有存在有效许可，则允许程序运行，否则不允许。加壳后的程序与原始程序的调用方式相同。

## 4.1 Windows程序加壳

在开发商套件中，可以启动外壳Envelope工具，如下图：

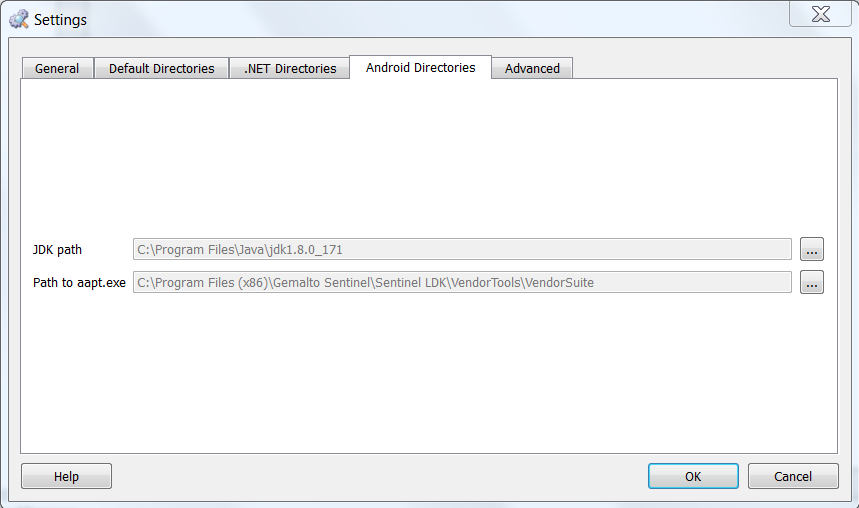


外壳使用非常简单，指定输入文件即原始二进制程序、输出文件即保护后的文件，然后点击保护（Protect）即可。最重要的设置是开发商代码（评估测试可以用DEMOMA）与特征 ID，特征ID要与生成许可时保持一致。如果使用正式开发号加密，请选择对应开发号。

对于Java与.Net程序，LDK外壳支持基于方法级的加密，即可以选择某个或某些方法（或函数）进行加密。

## 4.2 Android程序加壳

Android程序使用Windows外壳工具加壳，所不同的是设置JDK与aapt工具的路径（菜单File > Settings），如下图。



LDK外壳支持对apk文件以及.so库文件进行加壳，使用方法同Windows。如果对apk文件，加完壳需要先进行签名，然后在设备上安装运行。因为加壳后的应用需要有license（软锁或硬锁）才能运行，LDK对apk加壳时缺省会添加用于获得和升级license的代码（称为RUS），并将这部分代码放在初始activity，也就是说应用启动时如没有检测到license，会自动加载这个activity。

这部分详细说明请参见C:\Program Files (x86)\Gemalto Sentinel\Sentinel LDK\Additional Platforms\Android\Docs下的Sentinel LDK for Android\_Getting Started.html。

*注意：使用DEMOMA测试号不能加密原生库（.so），正式号可以。如果您还没有购买主锁，但需要测试.so加密，请发送该.so给我们，我们帮您加密。激活（只能是离线激活）时，把生成的设备指纹（.C2V文件）发给我们，我们把生成License（.V2C文件）再发给您安装使用。*

## 4.3 Linux程序加壳

Linux下没有图形界面的外壳工具，仅提供命令行工具（linuxenv）。如程序运行的目标机器是Linux ARM平台，加壳工作也需要在x86平台下进行，加完壳后在ARM平台下运行即可。

**步骤如下：**

1. 在Windows下安装授权管理服务器EMS与LDK开发商套件（开发商套件为可选）。
2. 解压ISO光盘安装文件，将Linux子目录下所有文件拷贝到目标Linux主机上的某个目录下如$HOME/Gemalto。
3. 安装运行环境（可选，仅适用与早期的有驱硬锁与AdminMode 软锁）。方法是在Gemalto/Redistribute/Runtime目录下运行：

* 对于Debian、Unbuntu：

dpkg -i aksusbd\_7.90-1\_i386.deb  
dpkg -i aksusbd\_7.90-1\_amd64.deb

* 对于Redhat、CentOS：

rpm -i aksusbd-7.90-1.i386.rpm （升级用-U）

rpm -i aksusbd-7.90-1.x86\_64.rpm

1. 在Linux下编译你的C语言代码。

*注意：如在代码里调用License API ，需在gcc选项里加相关选项（如-I –L 与-l）指明头文件及库文件（如libhasp\_linux\_x86\_64\_demo.a）的目录。如果只加外壳，则无需加这些编译选项。*

1. 用外壳工具linuxenv（在Gemalto/VendorTools/Envelope目录下）加密编译后的执行程序（假设为example）：

./linuxenv -v:../../VendorCodes/DEMOMA.hvc -f:100 example example\_pr

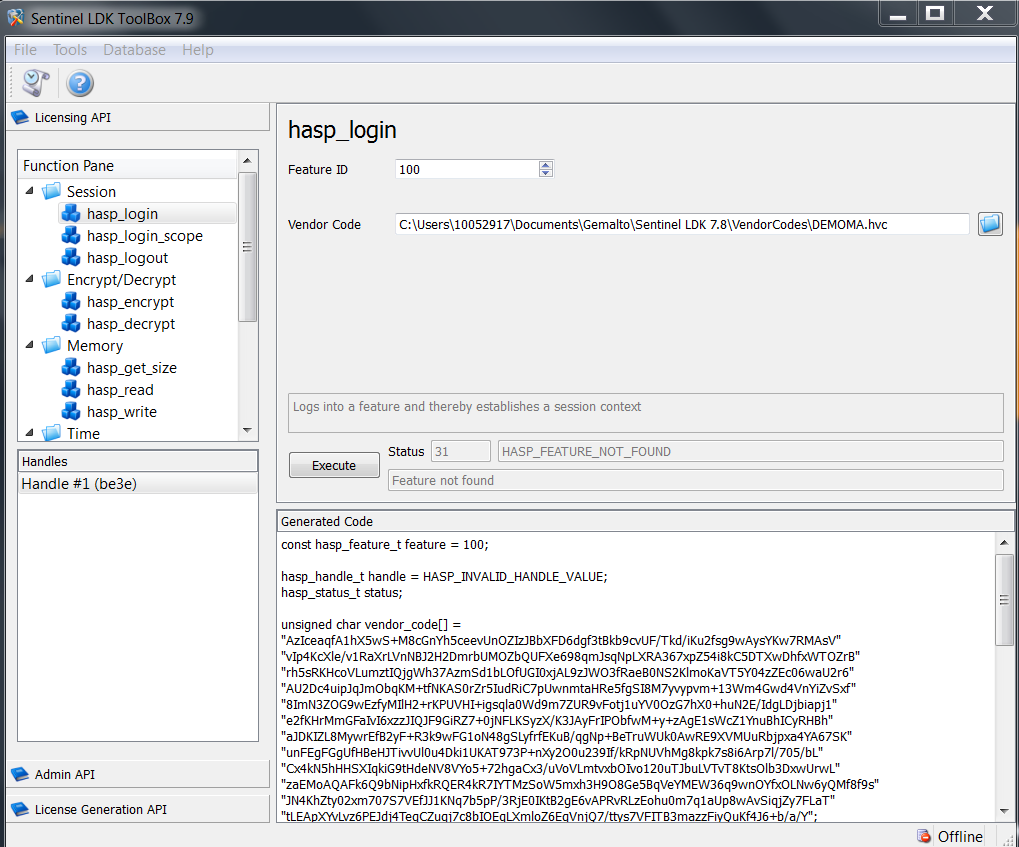
* -v：后的参数表示开发商代码VendorCode文件（如果是测试套件，则为DEMOMA.hvc）
* -f：后的参数是特征ID，这里为100
* --dfp: 如需打开加密的数据文件，需使用选项
* --debug: 关闭debug程序检查
* --memdup: 关闭内存转储
* --help：显示帮助信息
* 后面两个参数分别表示要加密的执行程序（example）以及加密后的执行程序（example\_pr）

# LDK API

通过在源代码里集成API，开发商可以实现基于特征的许可功能。LDK API支持多种操作系统（包括Windows、Linux、Mac、Android）与编程语言（包括C/CPP、.NET、Java等）。API方法命名简单明，通常以hasp\_开头，如

* 用于登入/登出授权的hasp\_login()/hasp\_logout()
* 用于数据加密/解密的hasp\_enrypt()/hasp\_decrypt()
* 用于读写锁上内存的hasp\_read()/hasp\_write()
* 用于获取机器指纹及授权信息的hasp\_get\_info()

在发商套件中有ToolBox工具（如下图），用于查看API如何使用并模拟API执行。



## 5.1 Licensing API

Licensing API主要提供许可相关的功能，如检查许可（即License）、数据加密等。Licensing API可以对程序的特定功能或某段代码进行单独的许可控制，实现按功能授权，而不是像外壳那样只能对整个程序进行授权。

下面是一个简单的使用Runtime API的例子。假如一个多功能打印机有打印（print）、复印（copy）与传真（fax）三个功能，其主程序代码如下：

void main()

{

print(); //打印

copy(); //复印

scan(); //扫描

}

如果打印、复印、传真对应的特征ID分别是11、12、13，则集成Licensing API后的代码为：

void main()

{

if (hasp\_login(11) ==HASP\_STATUS\_OK) { //特征11存在合法许可

print();

}

if (hasp\_login(12) == HASP\_STATUS\_OK) { //特征12存在合法许可

copy(); //复印

}

if (hasp\_login(13) == HASP\_STATUS\_OK) { //特征13存在合法许可

scan(); //扫描

}

}

注：hasp\_login()用于检查（或访问、登录）某个特征对应的许可，如合法有效，则返回成功。

LDK套件自带的各种语言的示例代码默认在C:\Program Files (x86)\Gemalto Sentinel\Sentinel LDK\Samples\Runtime目录下。

## 5.2 Admin API

客户除了可以使用ACC工具（即<http://localhost:1947>）管理本机或远程机器上的软许可和硬许可，也可以使用开发商提供的、用Admin API写程序实现类似的管理功能。

## 5.3 LicGen API

用于生成硬许可或软许可。开发商可以用EMS的图界面生成许可，也可以用我们提供的License Generation（LicGen）API自己写程序生成许可。 实际上EMS生成许可以也是调用了LicGen API。