

# VirboxProtector 1.0.3

北京深思数盾科技股份有限公司

2017. 4. 6

# 目录

1 概述	2
2 联系我们	2
3 产品导航页	2
4 产品主界面	2
4.1 菜单栏	3
4.1.1 打开	3
4.1.2 最近打开的文件	3
4.1.3 保存	3
4.1.4 立即加壳	4
4.1.5 登录云锁	4
4.2 导航栏	4
4.2.1 授权信息	4
4.2.2 被保护的函数列表	5
4.2.3 加密选项	6
4.2.4 消息选项	7
4.3 状态栏	7
4.4 可视化添加函数模块	7
4.5 如何使用sdk标签，添加需要保护的函数模块	10
5 问题及解决方法	11
5.1 什么是许可	11
5.2 许可获取以及许可的更新、删除操作	11
5.2.1 使用开发商工具签发许可	11
5.2.2 新建	11
5.2.3 编辑	11
5.2.4 删除	12
5.3 授权管理	12
5.3.1 新建模板	12
5.3.2 编辑模板	13
5.3.3 删除模板	13
5.3.4 发布许可	13
5.3.5 查看已签发许可	15
5.3.6 以 WEB 方式签发云许可	15
5.4 加壳后的程序在哪里	18
5.5 加密后的软件360报毒	18
5.6 加壳软件已知问题汇总	19
5.7 如何使用VirboxProtector_con.exe命令行加壳工具	19
5.8 如何生成map文件	20
5.9 如何对Unity3D程序进行加壳	23
5.10 如何对java程序进行加壳	25
5.10.1 对jar包进行加密	25
5.10.2 对war包进行加密	26

## 1 概述

深思数盾自动保护工具Virbox Protector，是深思数盾科技股份有限公司经过多年技术深耕开发的一款高强度自动保护（加密）工具。Virbox Protector与SS许可管理工具、深思云锁或精锐5或深思软锁配套使用，集自动代码移植、混淆、外壳加密、数据加密于一身，无需编程就能达到极高的保护强度，是业界领先的软件保护工具。

## 2 联系我们

深思数盾加壳工具技术支持QQ:2852513869



## 3 产品导航页



通过三种途径开始您的软件保护之旅：

- 将可需要保护的程序拖拽到窗口，进入加壳程序的主界面。
- 点击【新建】目录下面的【打开】按钮，打开需要保护的程序，进入加壳程序的主界面。
- 如果您已经在最近使用VirboxProtector保护过您的程序并保留该历史记录，您可以在【最近打开的工程】目录中找到您的程序并且单击打开程序，进入加壳程序的主界面。

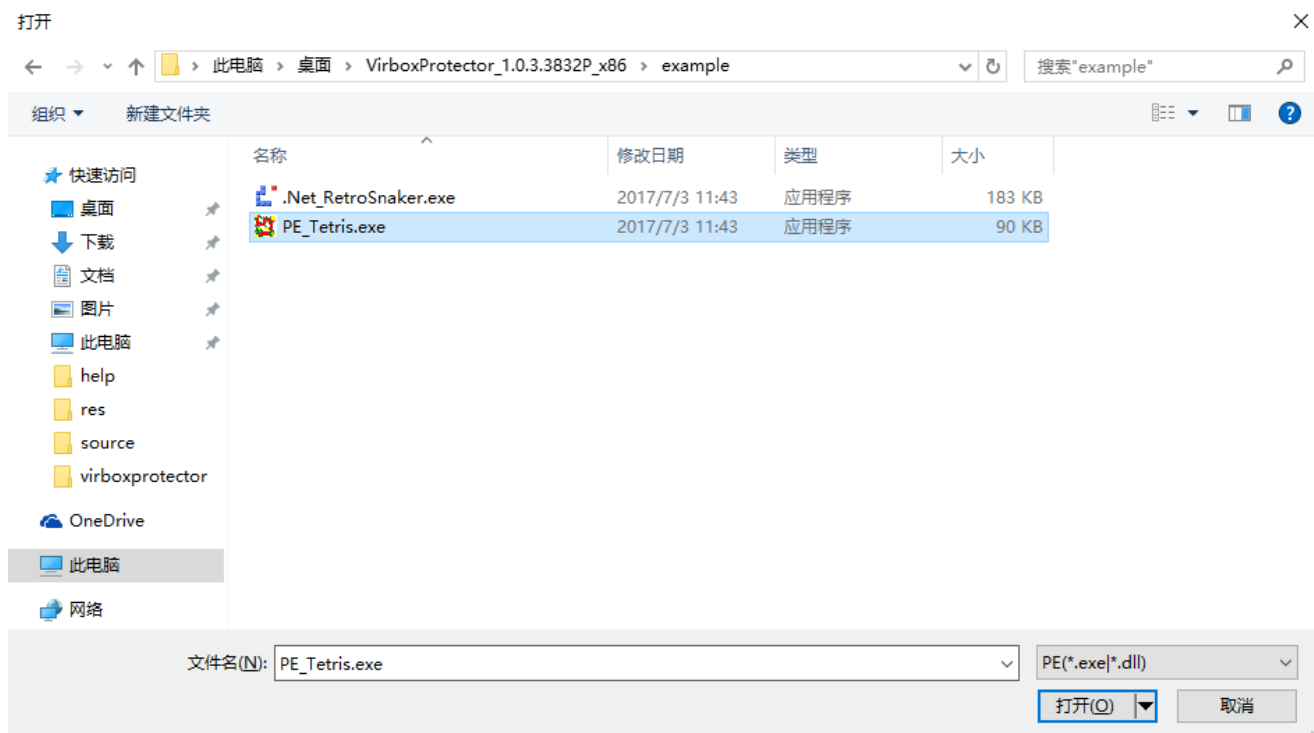
## 4 产品主界面



## 4.1 菜单栏

### 4.1.1 打开

- 点击【打开】按钮，打开一个需要加壳的程序，用户可以选择 XXX.exe或者是 XXX.dll，进入加壳程序的主界面。**[注意：如果被加壳程序的相同目录下存在XXX.map文件，那么会自动加载map文件，将函数名称显示在界面当中，目前支持vs、vc、bcb、delphi编译器生成的map文件]**



### 4.1.2 最近打开的文件

- 点击【最近打开的文件】按钮，系统罗列出5个最近保护过的程序，单击选择需要打开的程序，进入加壳程序的主界面。

### 4.1.3 保存

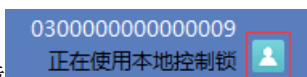
- 点击【保存】按钮，保存当前工程信息到 [需要保护程序名称.ssp]文件中。

#### 4.1.4 立即加壳



- 点击【立即加壳】即可绑定授权并生成保护后的应用程序.如果工程信息有所改动，那么点击【立即加壳】也会保存工程文件；如果工程信息未被修改，那么点击【立即加壳】不会保存工程文件。注意：[ 用户锁中必须存在和模块编号相同的许可ID才能够打开加壳之后的软件]

#### 4.1.5 登录云锁



- 点击【登录云锁】用户登录开发者账户，使用云控制锁进行加壳。如果不登录云控制锁，那么系统默认使用本地控制锁进行加壳，系统目前支持本地控制锁加壳和云控制锁加壳两种模式。**[特别注意：此处的云控制锁，登录的是开发者账户，而非普通的用户]**



## 4.2 导航栏

### 4.2.1 授权信息

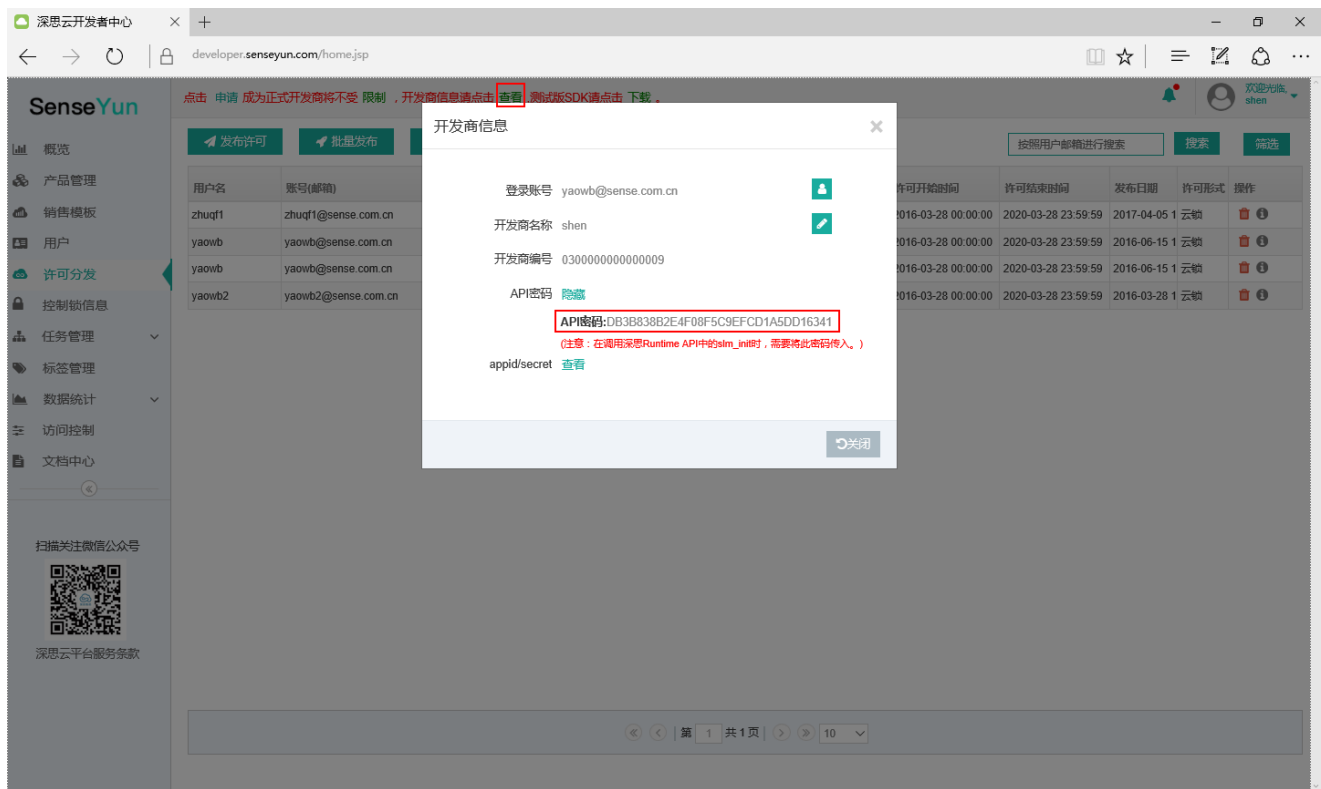
许可形式，保护后的软件可以使用的许可所在的载体。包括本地加密锁中的许可，表示插在用户本机器上的硬件锁；网络加密锁，插在局域网内其他机器上的硬件锁中的许可；云锁，云端虚拟的加密锁中的许可；软锁中的许可，软锁是用纯软件实现的加密锁，不依赖任何硬件锁，跟PC机器硬件唯一指纹绑定。

许可ID

开发者用于管理用户软件版权控制与发放凭证的一个序列号，开发者能够在开发者中心分发给普通的云账户，同时开发者也可以通过开发者管理工具为用户锁分发许可。

开发者密码

与开发者ID相对应的32个字符表示的序列号，保证了开发者的云授权平台的安全，每个开发者有一个唯一的开发者密码。开发者登录[深思云](#)获取。



## 锁序列号

一串32个字符表示的，能够唯一确定加密锁的序号，如果软件保护使用了指定的锁序列号，那么保护后的程序只能使用指定序列号的用户锁才能打开。

## 4.2.2 被保护的函数列表

需要保护的具有重要价值的函数块，用户能够选择混淆和碎片化代码的保护方式。

### 查看函数的详细信息

鼠标点击函数保护列表中的函数，在右侧的工作区窗口中展示函数的详细信息，包括函数的保护方式、函数名、函数的地址和汇编代码的展示。

类型	值
保护选项	
编译类型	混淆
详细信息	
名称	bounce2.Properties::Resources::get_ResourceManager
地址	402070
汇编代码展示	
40207C	ldsfld 0x04000001
402081	ldnull
402082	call 0x0a000018
402087	brfalse.s 32
402089	ldstr 0x70000001
40208E	ldtoken 0x02000003
402093	call 0x0a000019
402098	callvirt 0x0a00001a
40209D	newobj 0x0a00001b
4020A2	stloc.0
4020A3	ldloc.0
4020A4	stsfld 0x04000001
4020A9	ldsfld 0x04000001

## 函数的保护方式

用户能够选择不保护、混淆、碎片化、虚拟化四种保护方式。

序号	保护方式	说明	存在的问题
1	不保护	对选中的指令不进行保护	无
2	混淆	将代码指令翻译为机器和人都无法识别的一串伪代码字节流，在具体执行时在这些伪代码进行翻译解释，逐步还原为原始代码并执行	无
3	碎片化	将指令翻译成arm指令，并置于加密锁内执行	对指令有一定的格式要求，有的函数可能不能被保护
4	虚拟化	将指令编译为虚拟代码,放在指定虚拟机中运行	对指令有一定的格式要求，有的函数可能不能被保护

#### 添加保护函数的方法

- 1、使用SDK标签，编程方式添加需要保护的函数。
- 2、点击【添加】按钮，可视化选择添加保护函数。

#### 删除保护函数的方法

- 1、右键单击【被保护函数列表】->【清空列表】，即可实现保护列表框的清空操作。

- 2、将鼠标滑动至需要删除的函数名称上，点击【X】。

### 4.2.3 加密选项

#### 输出文件

输出文件，可以修改程序保护后生成文件的路径和名称。**[注意：1、如果只有文件名称，那么路径为源程序的路径；2、如果输出文件名和源文件同名，生成的程序会将源程序覆盖]**

#### 压缩

压缩，对加壳后的程序进行压缩处理，减小体积，同时可以防止静态反编译。**[特别说明：1、由于压缩模块需要一个固定大小的空间，如果被加壳的程序非常小压缩的效果并不明显还有可能出现体积更大的情况，对于体积较大的程序效果明显2、不支持DotNet动态库的压缩3、不支持arx类型程序的压缩]**

#### 名称混淆

名称混淆，对源程序中的函数名称进行混淆，静态反编译工具显示的函数名为乱码。**[特别说明：名称混淆目前只能支持DotNet程序，并且不支持IIS类型程序的混淆]**

#### 加密资源段

加密资源段，对被保护程序的资源区段进行加密，运行的时候需要用户使用相应许可进行解密方可使用程序。**[特别说明：加密资源段目前只能支持本地程序]**

#### 生成日志

生成virboxprotector的工作日志，方便出错的时候排查问题。正常情况下您可不选，只有在加壳后的程序运行出现问题的时候，您需要选择并将生成的日志发给深思技术人员进行排查问题。

#### 后台检测时间间隔(秒)

后台检测时间间隔（秒），表示每隔多少秒对运行程序进行检测是否存在对应许可，如果没有那么就会提示错误，或者退出。如果后台检测时间设为0s,那么后台就不会进行检测许可的操作。

## 4.2.4 消息选项

### 提示语言

设置弹出消息框文字的语言，包括简体中文、英文，繁体。

### 许可失效提示形式

许可失效提示形式，主要是用于设置，加壳后的程序找不到指定的许可的时候，如何进行提示。目前支持1、程序不提示消息，直接闪退；2、程序可继续运行，点击确定后退出；3、程序冻结，用户可选择重试连接许可或退出；4、允许找不到时，弹出登陆框；5、远程桌面服务会话消息框（支持vista和server2008以上版本）等五种不同的提示形式。**注意：**如果是对autocad的arx程序进行加壳，建议选择第五种许可会话失效提示形式。

### 延迟退出时间(s)

当许可会话已经失效时，会弹出提示框，如果此时选择了“取消”那么您的程序会在设定的延迟退出时间到达的时候退出，这样做的目的是给客户提供了存档的时间。

### 提示标题

设置弹出消息框的标题信息。

### 自定义信息

对主要的许可状态信息进行显示，开发者按照一定的格式自定义编辑提醒消息，使保护后的程序更加的人性化的向用户展示许可状态信息。

### 云锁登陆错误信息

主要是针对加壳后程序找不到许可的时候，许可失效提示形式选择了第四种，那么会弹出云账号登陆框，如果此时登录出错那么本程序会显示对应的错误消息。

### 错误信息

定义了对应的错误码显示的提示信息，错误信息的格式为【8位16进制错误码-错误信息】，可以添加错误消息也可以删除错误消息，如果没有自定义错误消息，那么加壳后的程序出现错误的时候会弹出系统错误信息。【详细的错误码介绍请参见SDK中的 `ss_error.h` 文件】

## 4.3 状态栏

---

状态栏中从左到右分别显示了被保护程序的文件的全路径、程序的版本、程序的类型以及程序的硬件机器版本。

## 4.4 可视化添加函数模块

---

添加保护函数

搜索

性能分析

可保护函数列表

名称	调用次数	保护方式	已添加
YXpen"(void)	--	不保护	
color"(void)	--	不保护	
AUTOTETRIS(int,int)	--	不保护	
Evaluate(bool **,int,int)	--	不保护	
WDtoWD(bool **,bool **,int *,int *,int ,int )	--	不保护	
amalqamate(bool **,bool **,int *,struct iPOINT,int ,int )	--	不保护	
bestDown(bool **,bool **,struct iPOINT)	--	不保护	
evaluateFK(bool **,int,int)	--	不保护	
init(bool **,int *,int,class FANGKUAI ,class FANGKUAI ,struct iPOINT	--	不保护	
sumBestPS(void)	--	不保护	
operator=(class FANGKUAI )	--	不保护	
Show(bool)	--	不保护	
change(int )	--	不保护	
getBarray(int ,int )	--	不保护	
initFK(int)	--	不保护	
setPoint(struct iPOINT )	--	不保护	
Show(bool)	--	不保护	

函数总个数 312; 已添加函数 1; 混淆个数 1; 碎片化数 0; 虚拟化数 0; 

添加

取消

函数搜索

在搜索框中输入关键字，程序会列出所有名称中包含关键字的函数块，并且彩色显示，支持模糊查询

添加保护函数

main

性能分析

可保护函数列表

名称	调用次数	保护方式	已添加
dllmain.obj DIIMain	--	不保护	
LIBCMT:dllcrt0.obj DIIMainCRTStartup	--	混淆	✓
LIBCMT:dllcrt0.obj DIIMainCRTStartup	--	不保护	
LIBCMT:dllcrt0.obj DIIMainCRTStartup\$filt\$0	--	不保护	

函数总个数 1775; 已添加函数 1; 混淆个数 1; 碎片化数 0; 虚拟化数 0; 

添加

取消

性能分析

点击性能分析按钮，运行需要保护的程序，执行正常的业务操作，然后关闭程序后，程序中各个函数模块调用的次数即可显示在列表中，参见上图。如果当前分析的程序位为I程序，需要选择启动模块。

可保护函数列表

展示了需要保护程序的所有的函数模块，托管代码程序和非托管代码程序有细微的差别。[注意：并不是所有的函数模块都能展示出来。1、指令大小小于15个字节的函数模块不会展示；2、有的非常规函数模块不会展示（名称中存在'!'、'<'、'>'、'@'、':'、'?'等）。]

托管代码程序

- 1、函数名称为“命名空间+类名称+函数名称”。
- 2、函数列表能够切换为列表的模式和树形模式，点击表头【已添加】右侧的按钮实现切换。

可保护函数列表

名称	调用次数	保护方式	已添加	
▼ bounce2	--			
▼ Program	--			
M Main	--	混淆	✓	
▼ Vector2	--			
op_Addition	--	不保护		
op_Subtraction	--	不保护		
op_Multiply	--	不保护		
BounceForm	--			
bounce2.Properties	--			

- 树形模式展示

可保护函数列表

名称	调用次数	保护方式	已添加	
▼ bounce2	--			
▼ Program	--			
M Main	--	混淆	✓	
▼ Vector2	--			
op_Addition	--	不保护		
op_Subtraction	--	不保护		
op_Multiply	--	不保护		
BounceForm	--			
bounce2.Properties	--			

- 列表模式展示，

可保护函数列表

名称	调用次数	保护方式	已添加	
M bounce2::Program::Main	--	混淆	✓	
op_bounce2.Properties::Resources::get_ResourceManager	--	不保护		
op_bounce2.Properties::Resources::get_h1d3_pow2	--	不保护		
op_bounce2.Properties::Resources::get_h1d3_pow3	--	不保护		
op_bounce2.Properties::Resources::get_h1d3_pow4	--	不保护		
op_bounce2.Properties::Resources::get_h1d3_yellowballp	--	不保护		
op_bounce2.Properties::Resources::get_magentaball	--	不保护		
op_bounce2.Properties::Resources::get_pow	--	不保护		
op_bounce2.Properties::Resources::get_powerball	--	不保护		
op_bounce2.Properties::Resources::get_yellowball	--	不保护		
op_bounce2::Vector2::op_Addition	--	不保护		
op_bounce2::Vector2::op_Subtraction	--	不保护		
op_bounce2::Vector2::op_Multiply	--	不保护		
op_bounce2::BounceForm::InitializeComponent	--	不保护		
op_bounce2::BounceForm::Collision	--	不保护		
op_bounce2::BounceForm::IsCollision	--	不保护		
op_bounce2::BounceForm::RenderPow	--	不保护		

列表模式中，点击列表头能够实现排序。

3、函数选择支持【Control + A】、【Control + 鼠标】和【Shift + 鼠标】等。

4、函数的保护方式支持右键选择

5、选择了保护方式后【已添加】会表示为已选择。

非托管代码程序

1、函数名称为函数的va的值

2、展示列表只有列表一种模式

其余的功能和托管代码程序一致。

### 信息展示

可保护函数列表信息展示，显示了函数的总个数、已添加函数个数、混淆函数个数、碎片代码函数个数。

函数总个数 17; 已添加函数 14; 混淆个数 11; 碎片化数 3; 虚拟化数 0;

添加

取消

### 添加

点击【添加】按钮，退出窗口，并将函数添加到主页面的被保护函数列表当中

### 取消

点击取消，退出窗口，不做任何修改。

## 4.5 如何使用sdk标签，添加需要保护的函数模块

sdk工具包，包括头文件、静态库以及动态库，用户在编程的过程中将sdk标签静态载入到需要保护的函数当中，这样生成的可执行程序，在Virbox Protector加壳工具中就能够分析出sdk表示的函数，这样就能够找到用户的核心代码所在的位置。目前支持，VBProtectBegin（常规保护），VBVirtualizeBegin（虚拟化保护），VBMutateBegin（混淆化保护），VBSnippetBegin（碎片化代码保护），VBProtectDecrypt(许可加解密)。[注意：sdk标签能方便用户找到关键代码]

sdk标签使用需要注意以下几点：

sdk标签使用

- 1、sdk只能静态加载，不支持动态加载dll（即loadlibraryA的方式）。
- 2、VBProtectBegin、VBVirtualizeBegin、VBSnippetBegin以及VBMutateBegin等接口，传入的字符串参数，不能与其他函数共用。
- 3、传入的字符串参数保证为ANSII码的形式，这样显示在界面上的函数名称才正确，否则就会显示为乱码。
- 4、每个Begin对应一个End，总是成对出现，并且一个函数里面不要出现多对begin+end。
- 5、如果sdk表示的保护方式和工程文件中保存的保护方式冲突了，以工程文件中的保护方式为准。
- 6、begin+end锁定代码最好大于3行代码。（因为锁定的代码正汇编生成的指令小于15个字节，那么不会显示在加壳工具界面上）
- 7、sdk动态库，分为32以及64位，在使用的时候要开发者根据要编译的程序位数进行加载对应的库。
- 8、目前明确不支持的语言：易语言、java程序、unity3d
- 9、begin/end不支持嵌套使用
- 10、VBProtectDecrypt被加密的字符串或者是缓冲区的长度必须是16的倍数，例如 `char g_test_string[16] = {"test_decrypt"};`
- 11、VBProtectDecrypt传入缓冲区和传出缓冲区不能是同一个缓冲区
- 12、VBProtectDecrypt传入的缓冲区只能放在函数外，即全局变量。具体的使用参照demo
- 13、.Net程序暂时不支持VBProtectDecrypt

## 5 问题及解决方法

### 5.1 什么是许可

LM本质就是「加密与管理」。深思许可管理（Sense License Mangment）是一套深思数盾根据中国国情、多年用户深度定制的经验与市场独特理解的许可管理解决方案，其本质是控制软件能否使用，尤其在领先的【移植算法】的理念保证下，解决软件无锁情况下试用、加密锁丢失甄别等棘手问题提供技术支持，达到超安全性与软件许可灵活性的平衡。同时针对企业内部挑战如加密开发难度高、加密方案泄露风险(加密人员离职)、私自烧狗等，提供强有力的工具与平台。许可是软件版权的控制与发放（software license），本质就是允许让最终用户能否使用软件的技术控制。本文主要讲的是技术控制，其中分为许可技术和许可管理。许可的技术控制：可以区分许可内容、许可使用时间、使用次数、软件并发限制、同时登录软件系统的在线人数限制等功能限制。许可管理：对许可使用、许可生命周期（产品创建、许可的生成、许可的发放、许可的吊销）的管理和用户的管理，综合为许可管理。

### 5.2 许可获取以及许可的更新、删除操作

#### 5.2.1 使用开发商工具签发许可

#### 5.2.2 新建

新建产品

产品名称: PE\_Tetris应用程序\*

产品版本: 1.0.0.1

关联许可ID: 1\*

许可版本: 0\*(0-4294967295的整数)

产品描述: PE\_Tetris测试程序

支持的授权方式: ☒加密锁

只读区: 当前大小: 0

设置大小

导入数据

☒锁定

读写区: 当前大小: 0

设置大小

导入数据

☒锁定

公开区: 当前大小: 0

设置大小

导入数据

☒锁定

编辑子模块区

确定

取消

#### 5.2.3 编辑

编辑产品

产品名称: PE\_Tetris应用程序\*

产品版本: 1.0.0.1

关联许可ID: 1\*

许可版本: 0\* (0-4294967295的整数)

产品描述: 仅用于测试

支持的授权方式: ☒加密锁

只读区: 当前大小: 0

设置大小 导入数据 ☒锁定

读写区: 当前大小: 0

设置大小 导入数据 ☒锁定

公开区: 当前大小: 0

设置大小 导入数据 ☒锁定

编辑子模块区

确定

取消

其中，锁定的意思是：选择后，新建模板及授权后，该数据区不可修改。

### 5.2.4 删除

选中要删除的产品，点击“删除”按钮。**注意：**删除产品时，会同时删除通过该产品创建的所有许可模板。


## 5.3 授权管理

授权管理包括模板的新建、编辑、删除功能，以及许可发布和已签发记录查看功能。

不同类型许可具有近似特征，重复操作一类许可不仅浪费时间，还会出现操作失误，不利于产品的管理和许可签发。深思使用模板的方式固化常用的许可类型，一次设定、反复使用，来减少开发者的工作量，提高效率。

### 5.3.1 新建模板

- 选择产品

 选择产品
 ×

产品总数: 1

产品名称	许可ID	产品描述	产品版本	许可版本	支持的授权方式
PE_Tetris应用程序	1	PE_Tetris测试程序	1.0.0.1	0	<input checked="" type="checkbox"/> 加密锁

1/1

- 填写授权信息

 新建模板
 ×

模板名称:  \*

许可类型:  ☒ 强制写入

产品:

☐ 永久许可

并发方式:

☒ 开始时间:

☒ 终止时间:

☒ 时间跨度:   天

☒ 使用计数:

☒ 只读区

☒ 读写区

☒ 公开区

☐ 子模块区

### 5.3.2 编辑模板

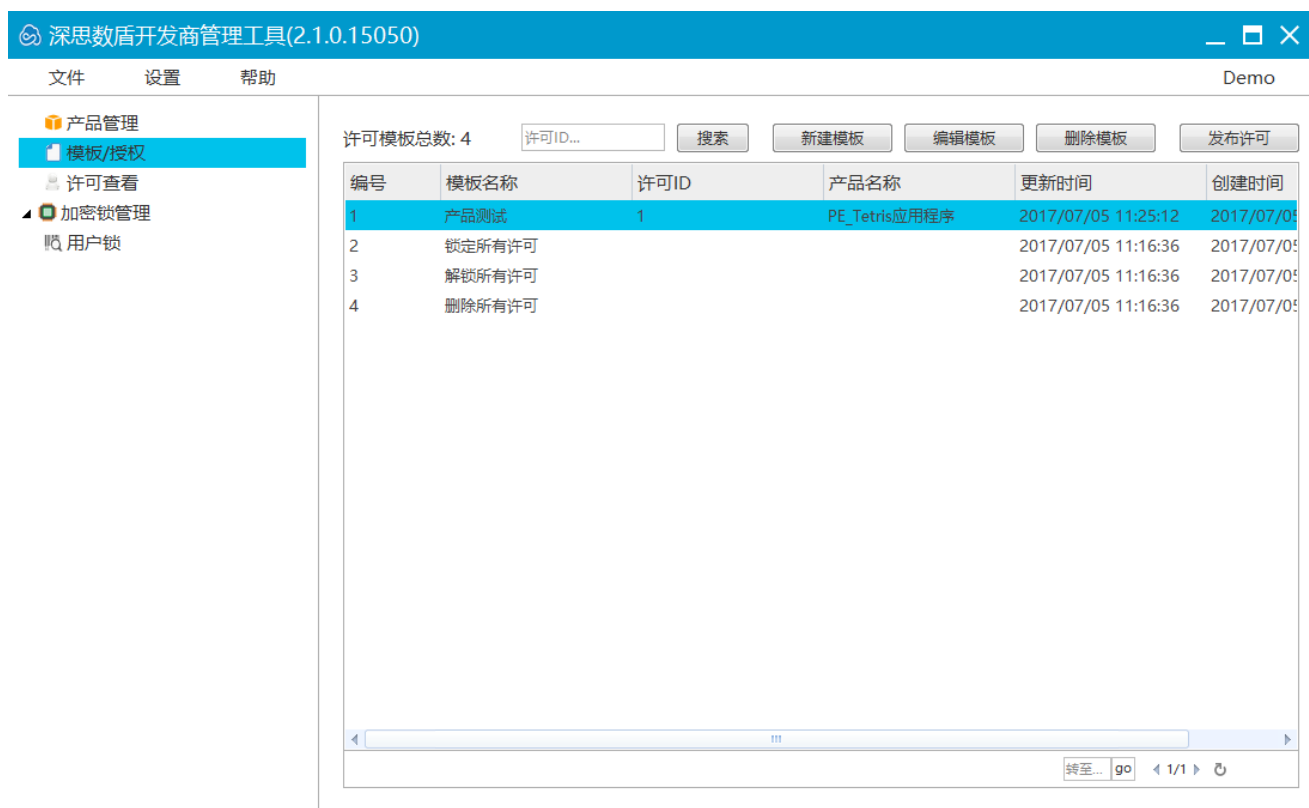
选择已有的许可模板，编辑模板信息。

### 5.3.3 删除模板

选中要删除模板，点击“删除模板”按钮。

### 5.3.4 发布许可

- 选中要发布许可模板，点击“发布许可”。



- 编辑授权信息

发布前可以对许可进行临时修改，适应于少量的改动的情况。

发布许可
×

模板编号:  \*      模板名称:  \*

许可类型: 添加 ☒ 强制写入

产品:

☐ 永久许可

开发方式: 单机

☒ 开始时间: 设置

☒ 终止时间: 设置

☒ 时间跨度: 设置  天

☒ 使用计数: 设置

☒ 只读区 编辑

☒ 读写区 编辑

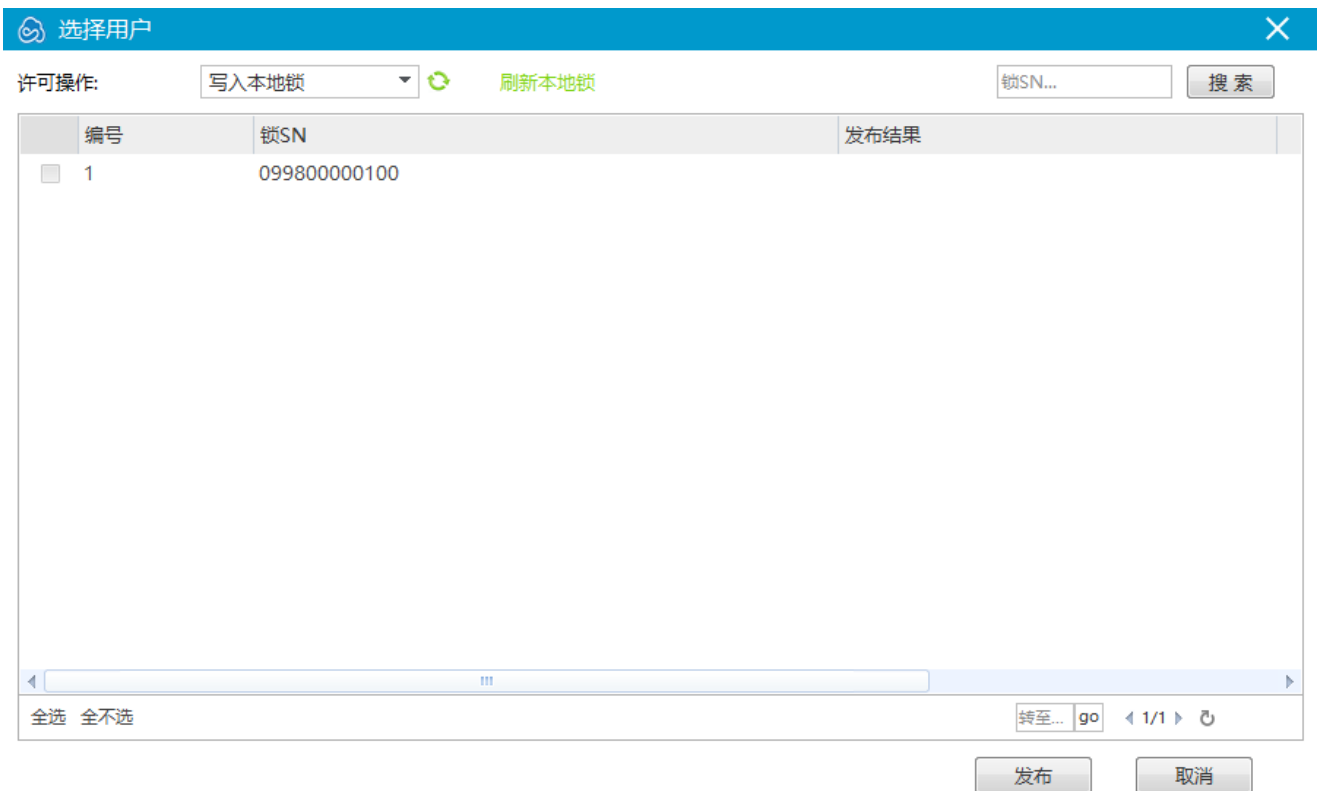
☒ 公开区 编辑

☐ 子模块区 编辑

下一步 取消

- 选择硬件锁

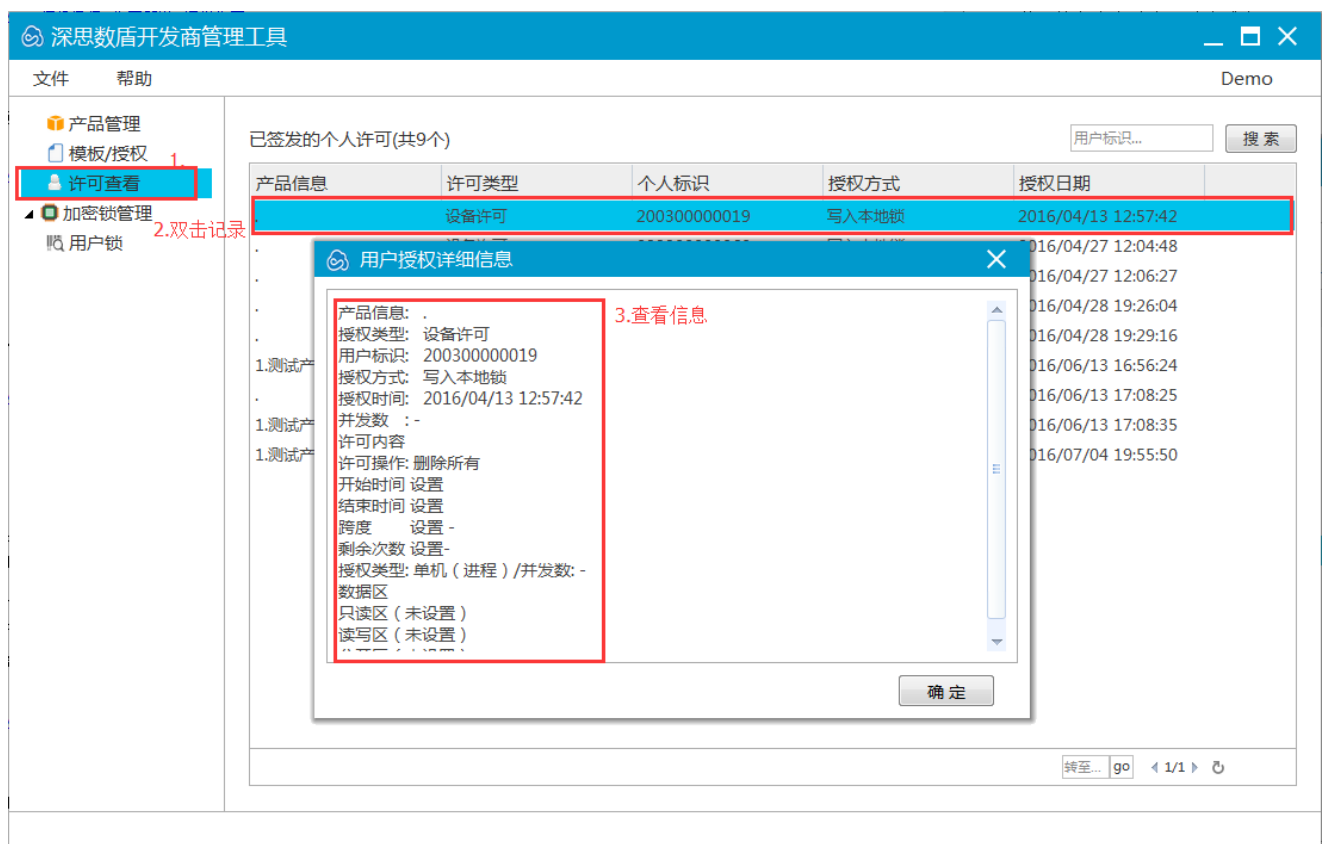
选择要签发许可的硬件锁，支持批量给多把锁签发许可。



- 许可签发 点击“发布”，发布成功。

### 5.3.5 查看已签发许可

查看已经签发的许可信息，如下图：



### 5.3.6 以 WEB 方式签发云许可

第一步 登录[开发者网站](#)的云账号

第二步 新建产品

深思云开发者中心

developer.senseyun.com/home.jsp

SenseYun

点击 [申请](#) 成为正式开发商将不受 限制 , 开发商信息请点击 [查看](#) ,测试版SDK请点击 [下载](#) .

按照产品模糊搜索 搜索

概览

产品管理

销售模板

用户

许可分发

控制锁信息

任务管理

标签管理

数据统计

访问控制

文档中心

扫描关注微信公众号

深思云平台服务条款

+ 新建

LicenceID	产品名称	创建时间	许可形式	操作
70	My Test	2016-03-28 13:11:50	云锁	

第 1 共 1 页 10

### 第三步 新建正式模板

深思云开发者中心

developer.senseyun.com/home.jsp

SenseYun

点击 [申请](#) 成为正式开发商将不受 限制 , 开发商信息请点击 [查看](#) ,测试版SDK请点击 [下载](#) .

按照模板名称搜索 搜索

+ 新建

产品管理

销售模板

用户

许可分发

控制锁信息

任务管理

标签管理

数据统计

访问控制

文档中心

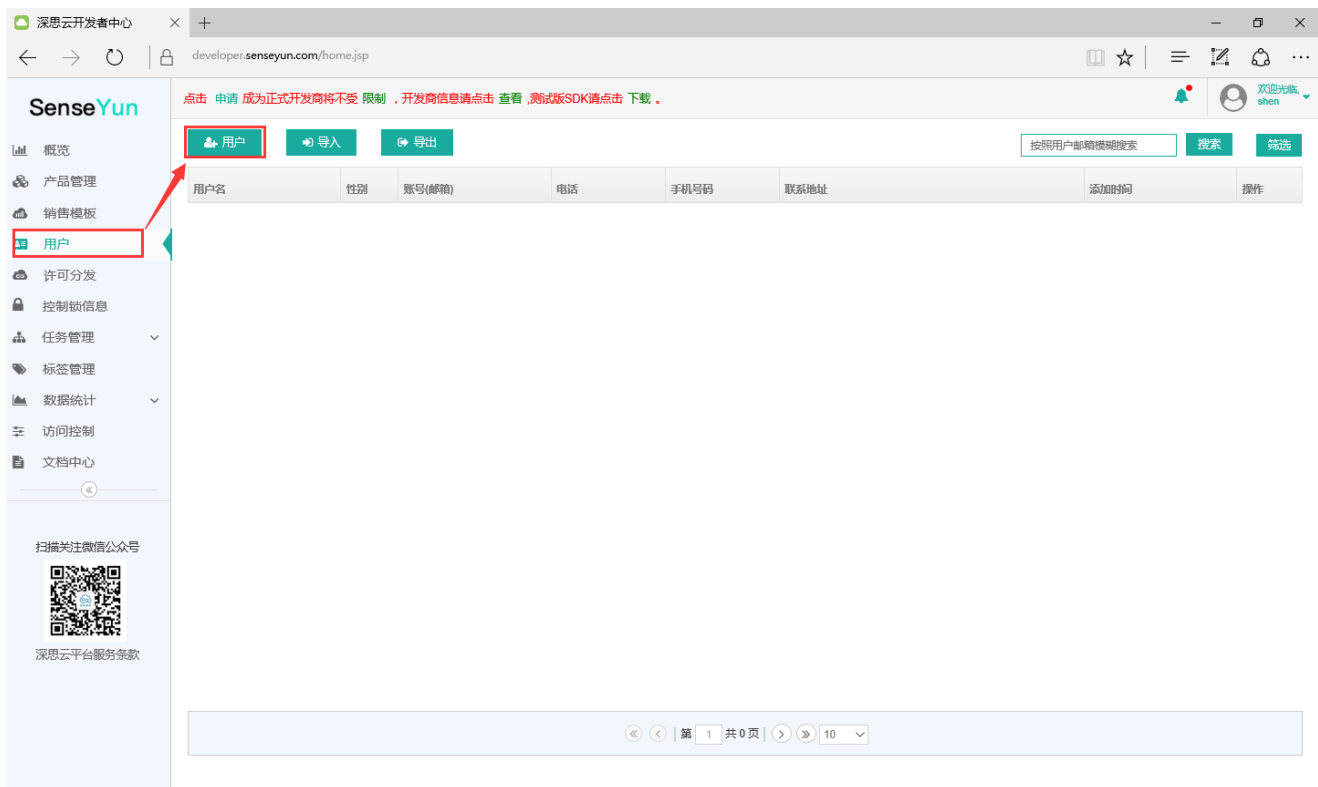
扫描关注微信公众号

深思云平台服务条款

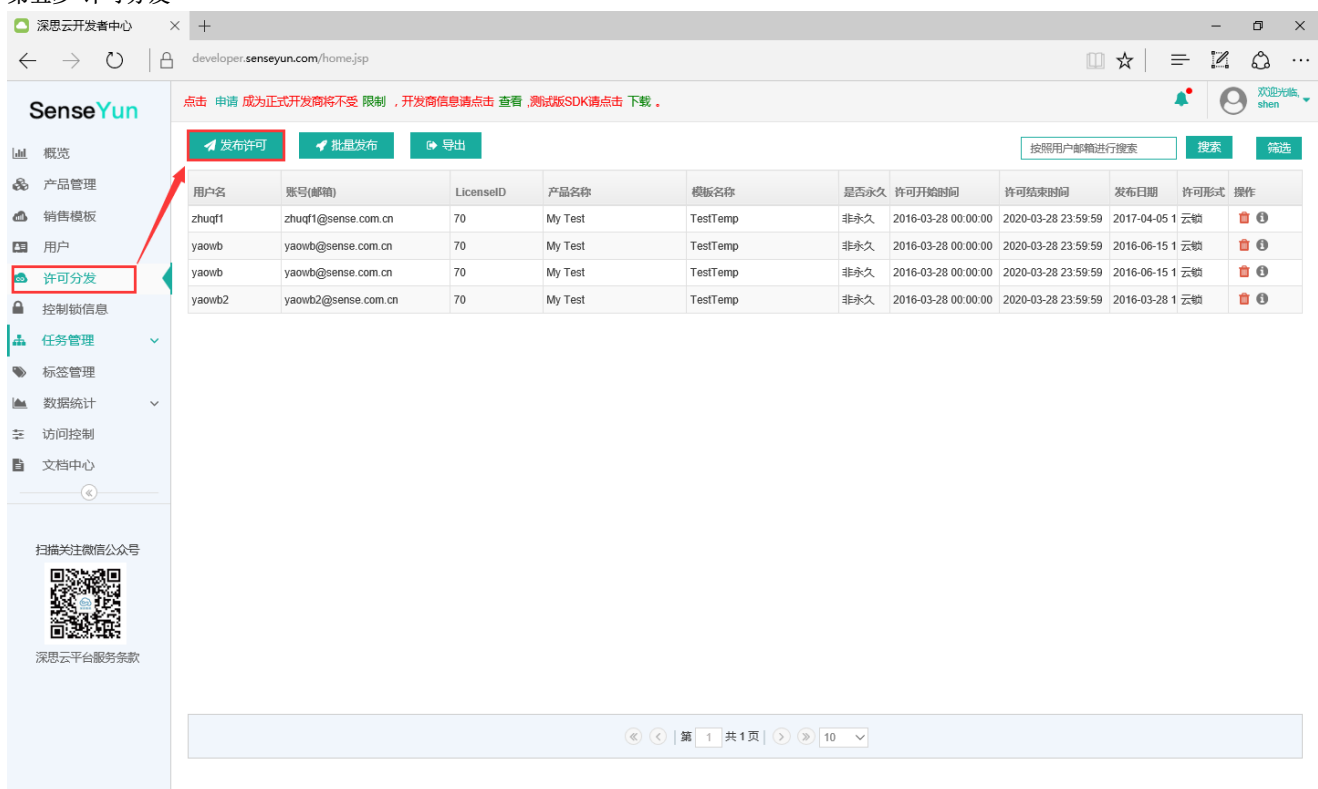
编号	模板名称	LicenceID	产品名称	创建时间	许可形式	操作
200	TestTemp	70	My Test	2016-03-28 13:40:00	云锁	

第 1 共 1 页 10

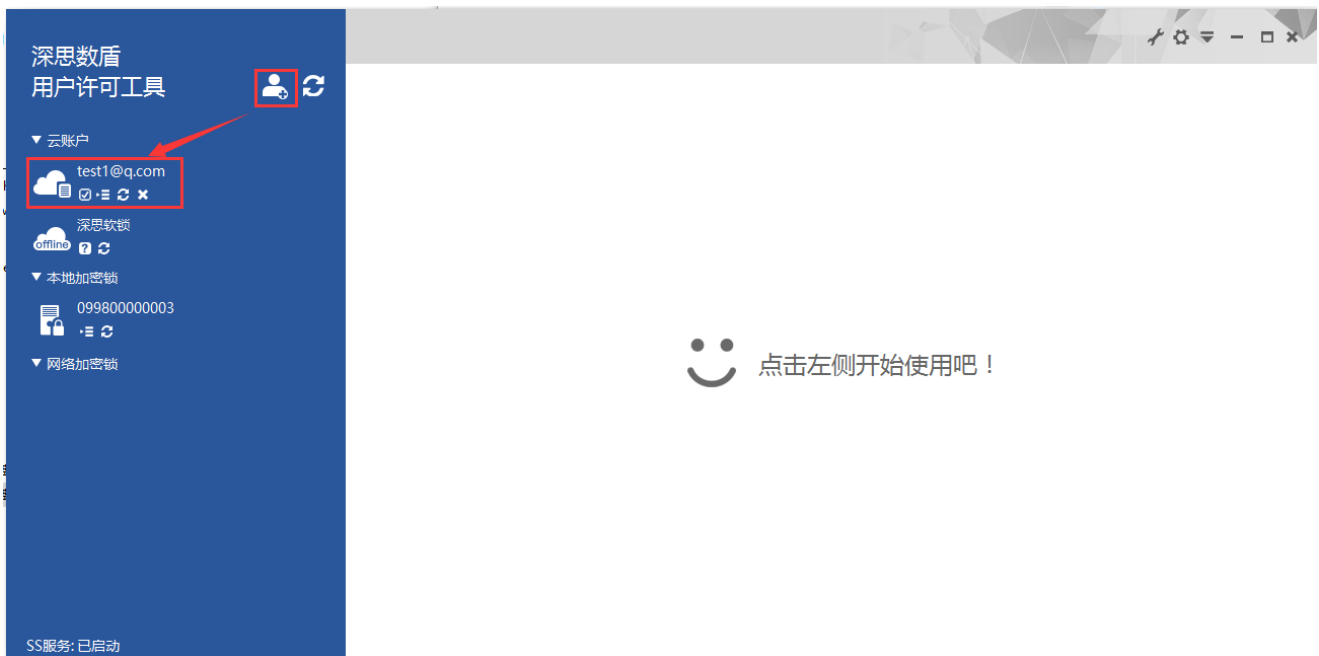
### 第四步 新建用户



## 第五步 许可分发



第五步 在许可管理工具中查看发布结果



## 5.4 加壳后的程序在哪里

问题描述：有一部分用户，找不到加过壳的程序，错误的将自己原始的程序当成加壳保护过的的程序。解决方法：我们以扫雷程序 Mine.exe 为例，加壳保护后的程序为 Mine.ssp.exe，与 Mine.exe 为同一目录下。

## 5.5 加密后的软件360报毒

问题描述：使用加壳工具对开发者软件进行加壳，然而加密后的程序360认为是病毒软件。

解决方法：提交360认证

第一步：软件中可执行文件（不包含驱动程序.sys）及打包后的可执行文件使用沃通（Wosign）签名（使用沃通代码签名工具并且购买代码签名证书进行签名，沃通签名相关问题请咨询沃通官方网站客服人员，沃通签名在360认证过程有很大帮助）。

第二步：注册并登录360开放平台：<http://open.soft.360.cn/>

第三步：在“我的软件”页面中选择提交我的软件，如下图：

请选择提交方式：

☒ 仅软件检测  
(快速通过软件检测，不收录到管家，防止误报)

☐ 软件检测并收录到管家 **推荐**  
(通过软件检测，并收录到国内最大下载平台360软件管家，获得有力推广)

请填写软件相关信息进行提交 ( \* 为必填项 )

软件名称:  \* 10个汉字内，英文字符不超过20个

软件版本:  \* 请输入数字或“.”

提交方式: \* 以下方式中任选一种即可

方式一: 提供官网下载URL, 易通过

方式二: 本地上传安装包

方式三: 批量上传

**上传须知:** 软件开放平台账号仅可以上传注册公司自主研发的软件，账号上传的任何软件都会视为公司行为；对于恶意上传病毒、木马的行为，开放平台将会关闭注册账号并协助网监和公安机关严肃追究法律责任。详细审核标准见：[平台介绍](#)

第四步：选择“仅安全检测”，并填写软件名称，软件版本（主要为了后续记录查看），提交方式可以选择本地上传安装包，然后提交软件。

第五步：同时可以在“软件列表”中查看已经提交过的待检测软件和“通过检测”的软件。目前软件提交后通过检测的时间大约为1天。

## 5.6 加壳软件已知问题汇总

- 不支持二次加壳，无论是第三方还是本程序加壳后的文件，都不能再次进行加壳。
- .NET加壳不支持第三方运行时库，只支持微软标准运行时库。
- 使用命令行加壳时，其目标程序的配置文件必须存在。
- 在.NET程序中使用类似GetField("name", bindingAttr)函数时，加壳后名称混淆可能出现异常，如果.NET 程序加壳后运行失败，尝试去掉名称混淆。
- sdk标签不支持含有本地代码的托管程序
- 对函数块进行碎片化代码保护的时候，存在不能成功保护的情况，主要原因是，碎片代码对指令的长度过小，指令可能不可移植，存在跳转等情况。
- 有的软件加壳后需要把名称改为原来的名称才能运行。(例如：程序Mine.exe，加壳后的程序为Mine.ssp.exe，假如程序Mine.ssp.exe不能成功启动，有可能需要将Mine.ssp.exe改为Mine.exe才能正常使用)
- 如果程序中存在附加数据，加壳后的程序可能会出现未知的问题。
- 部分unity 3D程序，如果许可失效时无法弹出消息框，在许可失效提示形式中选择“远程桌面服务会话消息框”即可正常弹出消息框。
- 暂不支持spring框架开发的java程序

## 5.7 如何使用VirboxProtector\_con.exe命令行加壳工具

- 1、首先在运行输入框中输入"cmd.exe"打开控制台命令窗口
- 2、输入 路径+VirboxProtector\_con.exe /? 可以查看帮助信息

```
C:\WINDOWS\system32\cmd.exe

C:\Users\admin\Desktop\VirboxProtector_1.0.3.3283\bin>virboxprotector_con.exe /?
SenseShield Virbox [version: 1.0.3.3283]
Copyright(c) 2016 SenseShield Technology Co., Ltd. All rights reserved.

VirboxProtector_con <filename> [-c local|cloud] [-u username] [-p password]
[-o output]
-c local|cloud      : login specific types of sense control lock.
-u username         : login cloud's username.
-p password         : login cloud's password.
-o output           : specific output filename.
-?                  : show help information.

C:\Users\admin\Desktop\VirboxProtector_1.0.3.3283\bin>_
```

## 命令行加壳工具help信息

```
VirboxProtector_con <filename> [-c local|cloud] [-u username] [-p password] [-o output]
-c local|cloud|auto          //login specific types of sense control lock.
-u username                  //login cloud's username.
-p password                  //login cloud's password.
-o output                    //specific output filename.
-?                            
```

### 3、使用本地控制锁加壳

VirboxProtector\_con.exe全路径 + 需要被保护的程序的全路径 -c local -o 输出文件的全路径。如图：

```
C:\Users\admin>C:\Users\admin\Desktop\setup\bin\virboxprotector_con.exe C:\Users\admin\Desktop\case\Mine.exe -local -o xxx.exe
SenseShield Virbox [version: 1.0.3.3074]
Copyright(c) 2016 SenseShield Technology Co., Ltd. All rights reserved.



loading Mine.exe ...
compiling...
save as xxx.exe ...
Succeed.
```

### 4、使用云控制锁加壳

VirboxProtector\_con.exe全路径 + 需要被保护的程序的全路径 -c cloud -u 开发者名称 -p 开发者密码 -o 输出文件的全路径。如图：

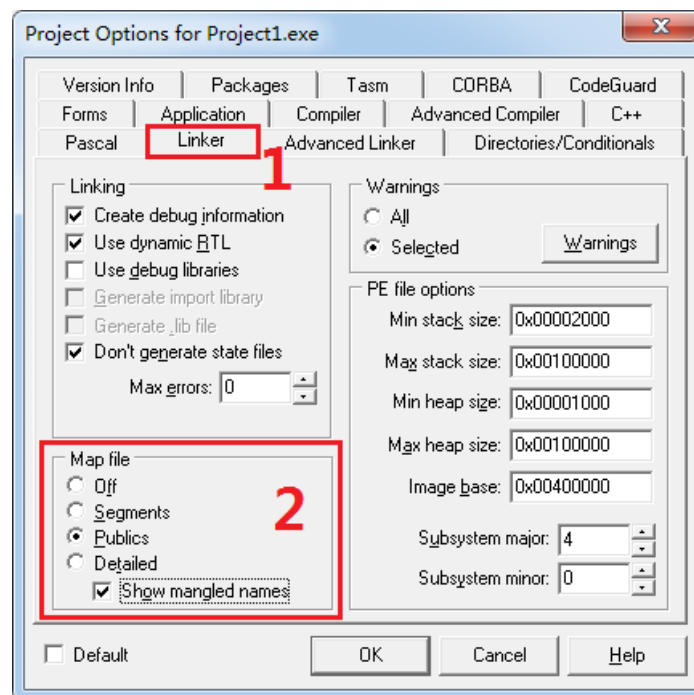
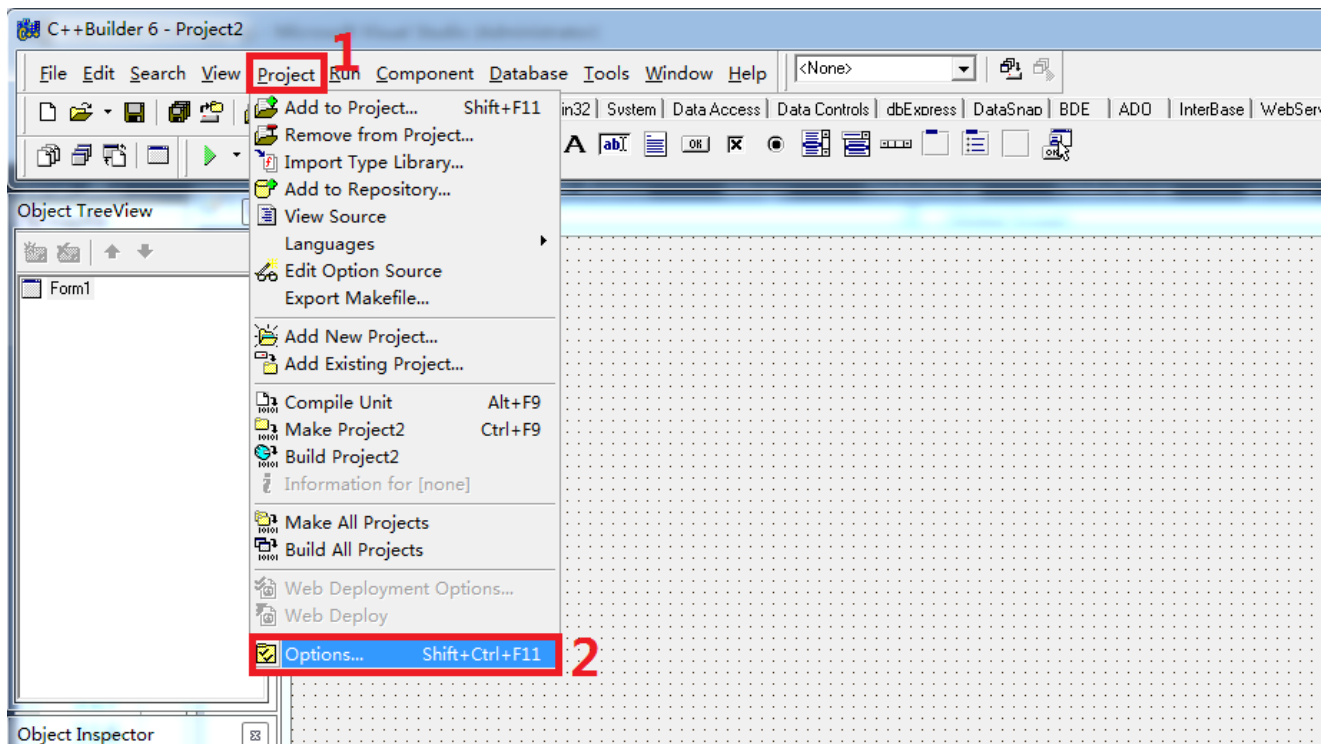
```
C:\Users\admin>C:\Users\admin\Desktop\setup\bin\virboxprotector_con.exe C:\Users\admin\Desktop\case\Mine.exe -cloud -u senseCloud06@sense.com -p XXXXXX -o xxx.exe_
```

注意：1、命令行工具会自动从被加壳工具下面查找 XXX.exe.ssp工程配置文件（该配置文件是Virbox Protector界面工具生成得到的）；2、如果路径中存在空格需要将路径使用双引号("路径")括起来。

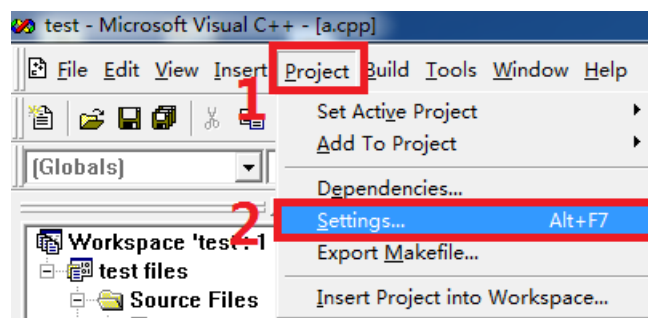
 Mine.EXE	2016/3/9 18:19	应用程序	224 KB
 Mine.EXE.ssp	2017/3/22 11:14	SSP 文件	3 KB

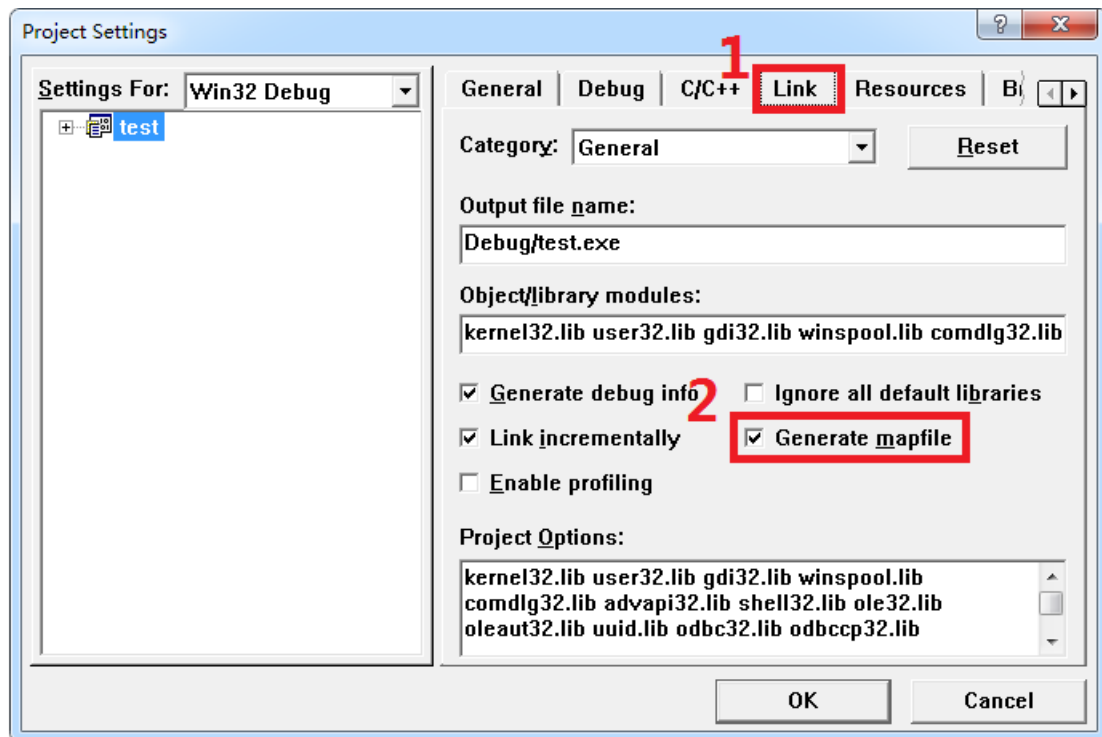
## 5.8 如何生成map文件

1、生成BCB程序map，工程设置如下图

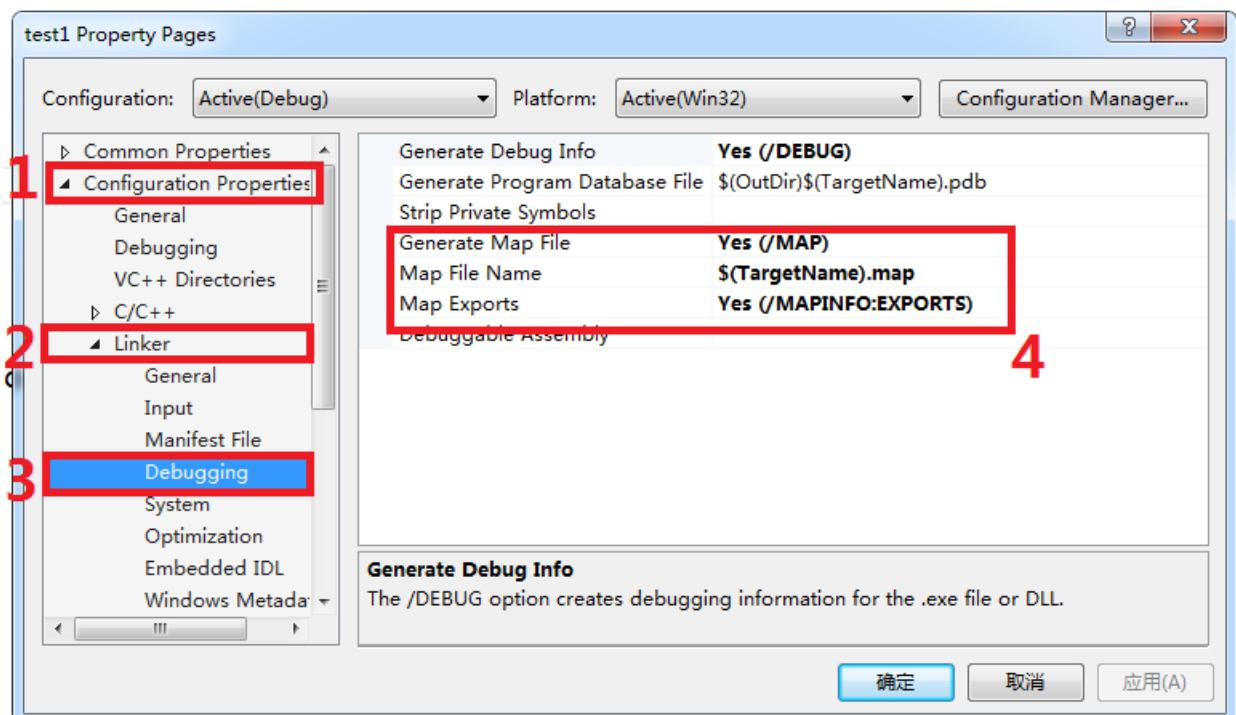
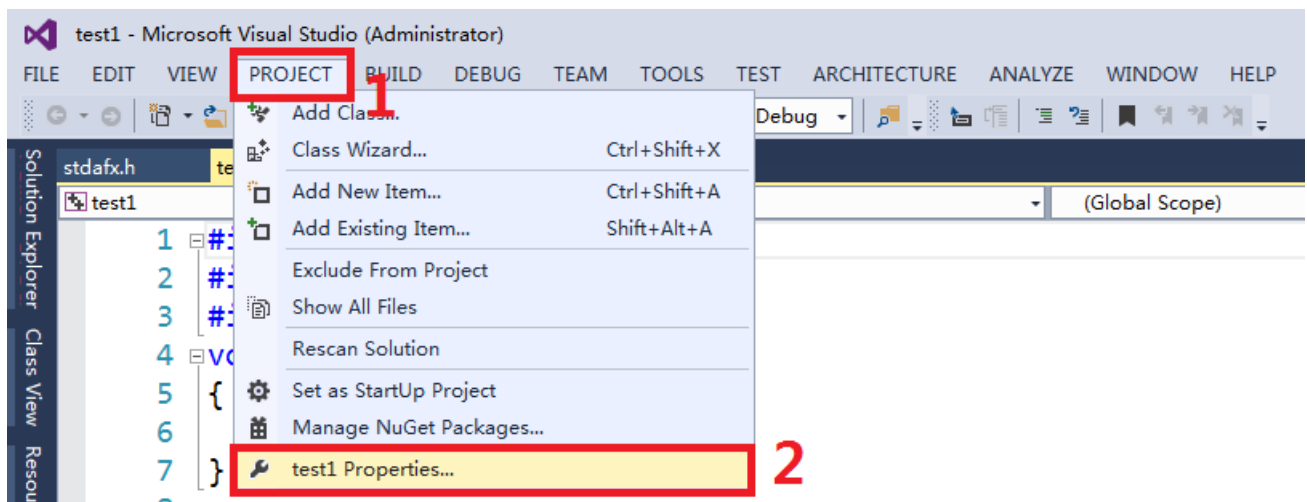


2、生成vc程序map，工程设置如下图

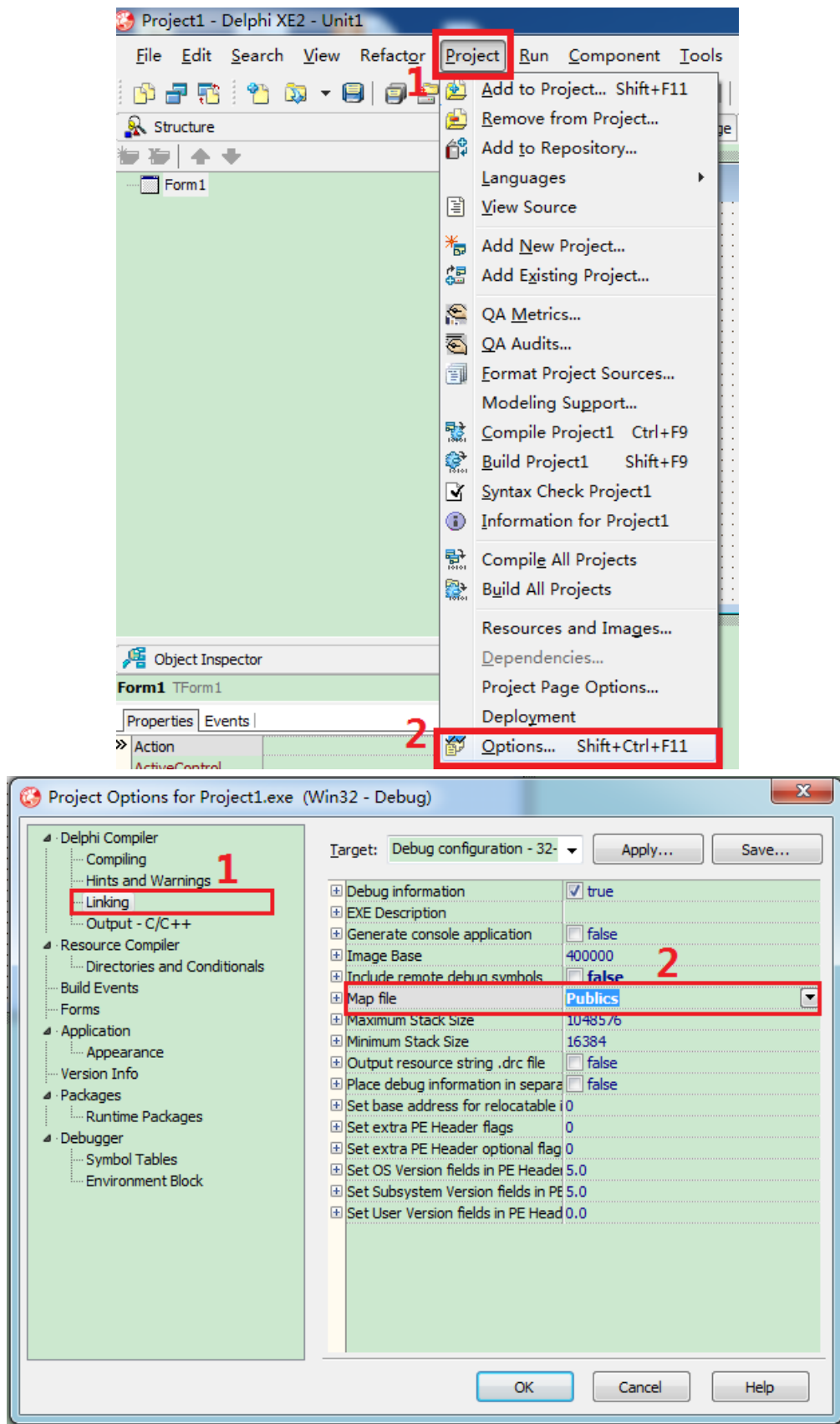




3、生成VS程序map，工程设置如下图

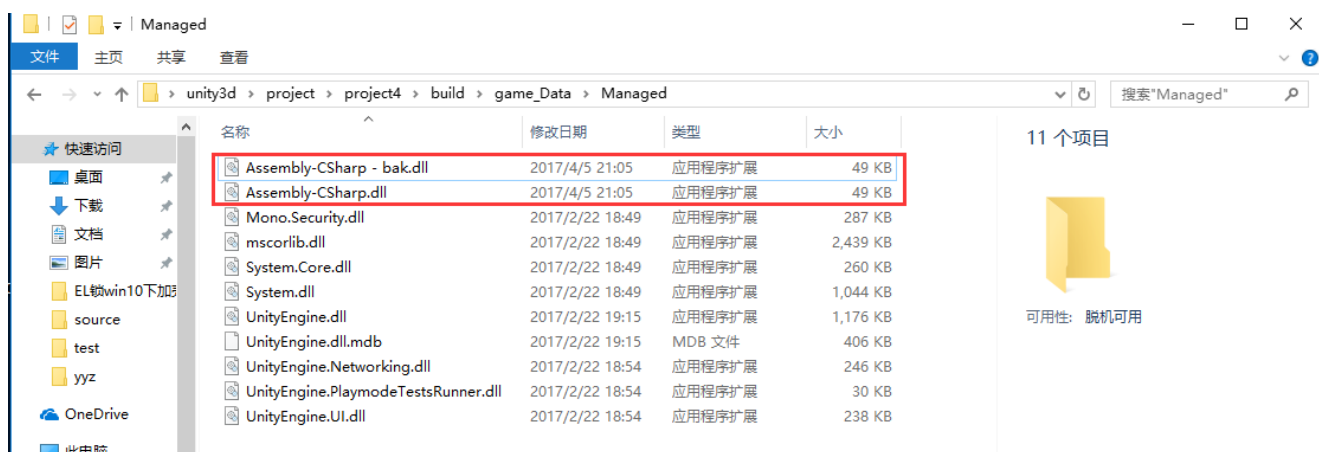


4、生成delphi程序的map，工程设置如下图



## 5.9 如何对Unity3D程序进行加壳

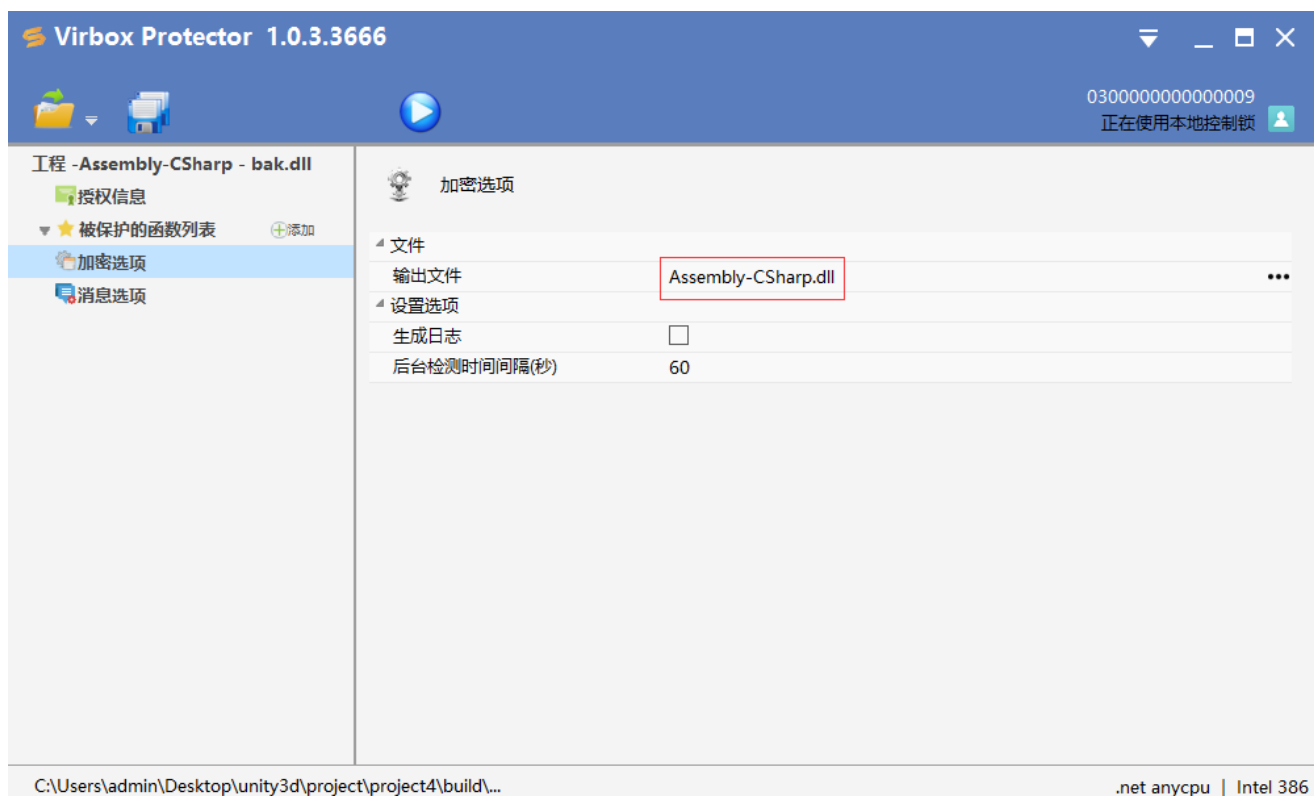
1、首先将您含有代码资源的文件修改名称。例如将Assembly-CSharp.dll改为Assembly-CSharp - bak.dll，因为加壳后的文件要改为原名称才能使用，所以需要将源文件改名备份，防止将您的源文件破坏。[注意：您的代码资源文件的上级目录下面许存在mono目录，同时mono目录下面须存在mono.dll文件。目前我们需要固定的文件目录结构，在文件加密的时候需要找到mono.dll进行处理]



2、将Assembly-CSharp - bak.dll文件拖入到我们的程序当中。



3、修改输出文件的名称。将【加密选项】->【输出文件】中的输出文件名称改为您程序原来的名称。因为我们的程序默认输出类似Assembly-CSharp - bak.ssp.dll的文件名，需要您手动改为Assembly-CSharp.dll（程序原来的名字）。



4、点击立即加壳。

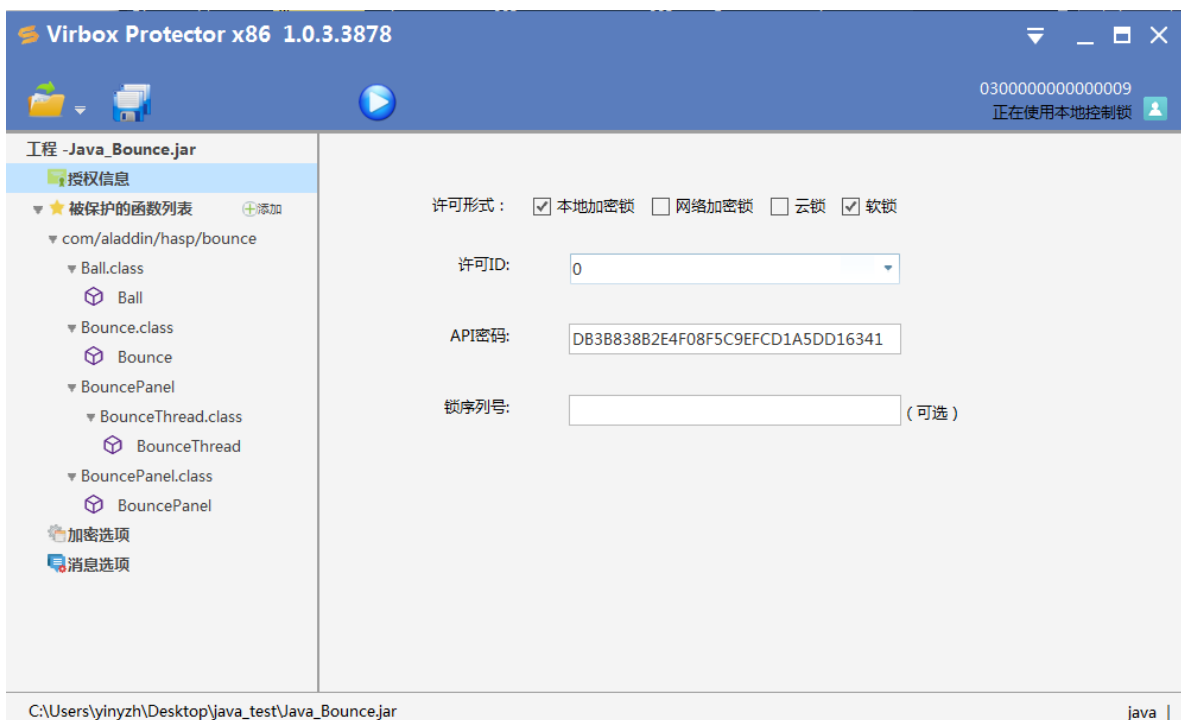


5、完成加壳。

## 5.10 如何对java程序进行加壳

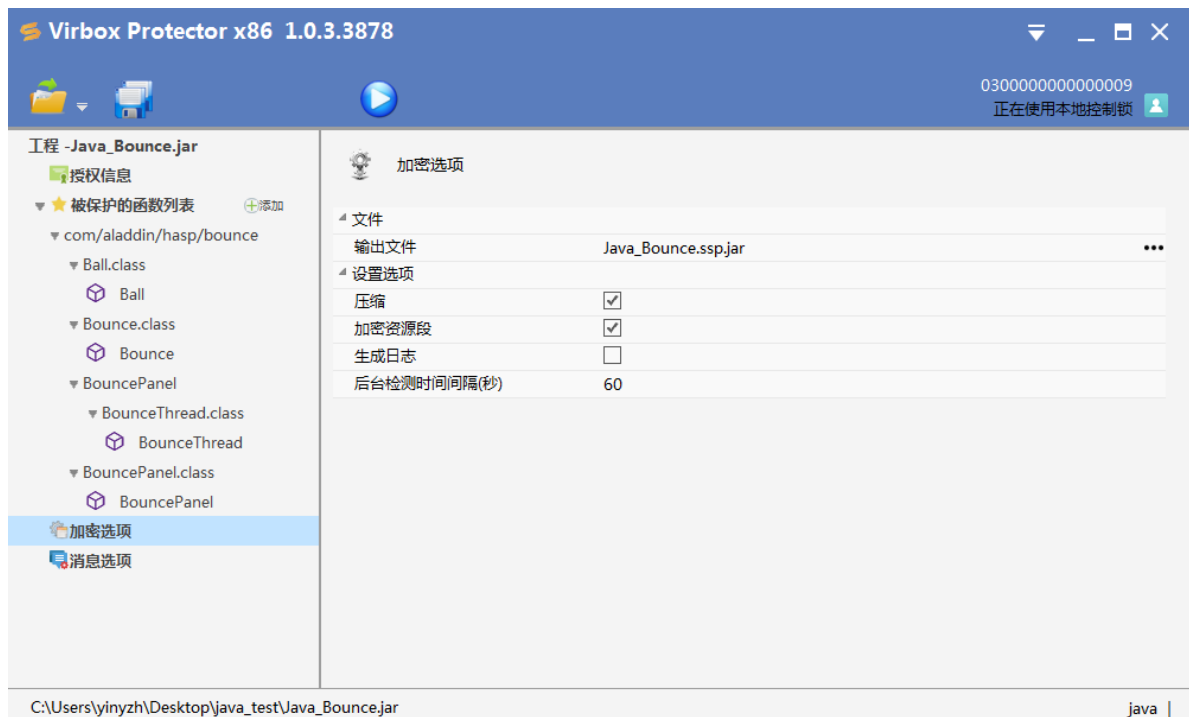
### 5.10.1 对jar包进行加密

第一步：将jar包拖入加壳程序



首页

第二步：设置加密选项



加密选项

第三步：设置消息选项

消息选项可以选择默认，也可以根据自己需要重新设置

第四步：点击【加壳】，生成加壳后的程序

Java_Bounce.jar	2017-04-17 21:02	Executable Jar File	5 KB
Java_Bounce.jar.ssp	2017-07-05 15:45	SSP 文件	4 KB
Java_Bounce.ssp.jar	2017-07-05 15:45	Executable Jar File	6 KB
sjt32.dll	2017-07-05 15:45	应用程序扩展	2,661 KB
sjt64.dll	2017-07-05 15:45	应用程序扩展	4,767 KB

加壳后的文件

第五步：运行加壳后的程序

命令行中输入：javaw -agentpath:sjt64.dll -jar Java\_Bounce.ssp.jar

```
C:\Users\yinyzh\Desktop\java_test>javaw -agentpath:sjt64.dll -jar Java_Bounce.ssp.jar
```

运行

## 5.10.2 对war包进行加密

第一步到第四步：war包和jar包加壳的流程是一致的。

需要注意：对war包进行加壳的时候消息选项中许可失效提示形式只能选择“程序不提示消息，直接闪退”。

第五步：运行加壳后的程序

- 1、需要将sjt32.dll或者是sjt64.dll【具体根据装的java环境决定】拷贝到jdk\bin\jre\bin中。
- 2、将set JAVA\_OPTS=-agentpath:sjt64.dll或者是set JAVA\_OPTS=-agentpath:sjt32.dll设置到tomcat中bin目录下的catalina.bat文件中
- 3、运行tomcat