

# 软件中心收费软件 **License** 保护编程指南

## <V1.1>

### 目录

1. 总体说明.....	2
1.1 阅读用户.....	2
2. 软件中心 <b>License</b> 保护.....	3
2.1 <b>License</b> 原理.....	3
2.2 实现方法.....	3
2.3 收费软件提供试用.....	5
2.4 插件 <b>License</b> 检查.....	5
2.5 注意事项.....	5
3. 参考资料.....	5

1. 总体说明

1.1 阅读用户

角色	描述
开发者	第三方收费应用开发者

1.2 修改历史

时间	修改内容
2010-5-13	增加试用软件的内容。

## 2. 软件中心 License 保护

### 2.1 License 原理

软件中心 License 保护,采用在程序中添加检查代码的方法来保护软件版权。

具体流程如下:

1、开发者在程序中添加 License 检查代码。

2、上传软件到软件中心，软件中心提取可执行软件的特征码

用户下载时，根据软件的特征码，以及用户机器 SN 等信息计算 License 传给用户机器，用户以 CAB 包中可执行文件部分特征的 MD5 作为文件名保存 License 文件在本地的 "Disk/Application Data/MyStore/Data/[MD5].license"

3、系统启动受保护程序时，读取 License 文件，并验证 License 文件的合法性以及 License 授权期间的正确性。

### 2.2 实现方法

我们提供一个 MyStoreLib 的静态库，

头文件：MyStoreLib.h

静态库文件：MyStoreLib.lib

头文件里面定义了一个 struct，表示 License 验证后的结果以及 License 包含的信息：

```
typedef struct stMYSTORE_VERIFY_CONTEXT
{
    DWORD Ver;           //< 防盗版版本号
    DWORD LicenseValid;  //< 证书是否合法（非法，：合法）
    DWORD Expired;       //< 是否过期（：未过期，：过期）
    DWORD ExpireDate;    //< 过期日期(如年月日：20100128)
    BYTE Reserved[19];   //< 保留字段
} MYSTORE_VERIFY_CONTEXT;
```

Ver 防盗版方案版本号，可以忽略。

LicenseValid, 表示 License 是否是发放给当前用户、当前软件的。

Expired, 如果是试用期软件，是否过期<软件中心目前不支持试用期，一律为正式软件>。

ExpireDate，过期时间。

Reserved 一些保留内容，目前仅用了前两个 bit。第一个 bit 置 1 表示收费，否则为免费；第二个 bit, 置 1 表示试用期，否则为正式 License。

License 验证的函数如下：

```
DWORD MyStoreVerify(const wchar_t* pszFileName, __out MYSTORE_VERIFY_CONTEXT*
pmyStoreVerifyContext);
```

pszFileName 为可执行文件的文件名称，可使用 GetModuleFileName 函数获取，pmyStoreVerifyContext 为验证结果保存的结构指针。

详细使用方法如下例：

```
#include "MyStoreLib.h"
#pragma comment(lib, "MyStoreLib.lib")
#pragma comment(lib, "PlatformAPI.lib")
#pragma comment(lib, "PhoneAdapter.lib")

wchar_t pszFileName[MAX_PATH] = {0};
GetModuleFileName(0, pszFileName, MAX_PATH);           //当前进程 EXE 的文件名
MYSTORE_VERIFY_CONTEXT mystore = {0};
DWORD dwRet = MyStoreVerify(pszFileName, &mystore);    //验证此文件是否合法
switch(dwRet)
{
case 0:
    {
        //验证结果为:合法
        // 检验是否超过试用期
        // 决定让应用程序继续运行
        RETAILMSG(1, (L"验证结果为:合法, LicenseValid:%u, Expired:%u,试用期终止日期:%u\n",
            mystore.LicenseValid, mystore.Expired, mystore.ExpireDate ));
        if(mystore.Expired)
        {
            //超过试用期
            //认为是试用软件做限制功能...
        }
        else
        {
            if(mystore.Reserved[0]&0x40 == 0x40)
            {
                //License 类别为免费 免费 License， 做功能限制。
            }else{
                //未超过试用期
                //认为是正式 License。
            }
        }
    }
    break;
case 1:
    {
        //验证结果为:打开 pszFileName 文件失败,
        // 认为是试用软件做限制功能...
        RETAILMSG(1, (L"验证结果为:打开 pszFileName 文件失败\n"));
    }
    break;
case 2:
    {
```

```
//验证结果为:打开 License 文件失败
// 认为是试用软件做限制功能...
RETAILMSG(1, (L"验证结果为:打开 License 文件失败\n"));
}
break;
case 3:
{
    //验证结果为:验证失败
    // 认为是试用软件做限制功能...
    RETAILMSG(1, (L"验证结果为:验证失败\n"));
}
break;
case 4:
{
    //验证结果为:序列号获取失败
    // ...退出应用程序
    RETAILMSG(1, (L"验证结果为:序列号获取失败\n"));
}
break;
}
```

## 2.3 收费软件提供试用

之前要求收费软件必须提供 Demo，供用户试用。现在提供新的试用策略，开发者不需要提交两个不同的软件，在软件验证 License 时，如果发现 License 文件不存在，或者验证失败，或者试用超时，或者 License 为试用版 License，那么进行功能限制。

如果您开发的是收费软件，那么如果检查 License 失败，则**仍然需要能启动程序**，但是程序是一个有功能限制的**试用版本**。

检查 License 失败的情况包括：

- 1、License 文件不存在
- 2、License 文件非法
- 3、License 文件合法，但是是试用期版本
- 4、License 文件合法，但是已过试用期

另外，提供试用版的收费软件，请在软件中心把软件的最新上架版本的允许使用选项打开。目前下载试用版软件不提供试用 License，所以验证试用软件时，一般结果是找不到 License。

## 2.4 插件 License 检查

如果开发的程序是桌面插件，那么代码里：

```
DWORD dwRet = MyStoreVerify(pszFileName, &mystore);
```

这一步的 pszFileName 文件名请指向 inf 文件里面注册表项定义的文件，如下的 abc.dll 文件：

"SOFTWARE\Meizu\MiniOneShell\Main\mCalculator", "ExecFileName", 0x00000000, "%InstallDir%\abc.dll"

## **2.5 注意事项**

- 1、对 License 的检查必须在桌面图标指向的可执行文件里面实现。
- 2、可执行文件不要太大，由于 License 的检查，可执行文件大小会影响程序的启动速度。
- 3、如果对一处 License 检查不放心，可以分多处调用。

## **3. 参考资料**

无。