

# 谈网页上 ActiveX 组件的打包 (Packaging) 和授权 (Licensing)

杨 强 张晓宁

**摘 要** 在 WEB 页面内使用 ActiveX 组件存在着如何在客户浏览器上顺利下载这些组件以及能否对它们进行运行时刻授权这些问题。本文旨在论述 CAB 文件和 LPK 文件是如何解决 ActiveX 组件的打包和运行时刻授权的。并对 WEB 程序设计者提供了范例。

**主题词** 打包 授权 CAB LPK

## 一、引言

利用组件 (Component) 创建桌面和网络应用程序已经成为当前软件设计的主流, 在 WEB 网页上程序员经常使用 ActiveX 组件来加强交互性能。由于普通应用程序的开发环境一般都提供 Setup Wizard, 因此在为最后生成的程序创建安装盘集的时候, 开发环境透明地实现了对组件的打包和授权 (授权分为 Design Time Licensing 和 Run Time Licensing 两种, 这里是 Run Time Licensing)。安装盘集一旦在用户机器上安装完成, 即可运行相应的应用程序, 程序设计员通常不会遇上程序中的组件在用户机器上不能使用这类问题。

与普通应用程序设计所不同的是, WEB 网页是一种超文本文档, 它可以由任何文本编辑器编写。一旦在 WEB 服务器上发布这些网页, 程序员就希望能在用户机器浏览器上查看到正确的输出结果。但如果在网页内嵌入了 ActiveX 组件, 在用户机器的浏览器上常常就会出现: 不是组件找不到, 就是组件下载了却不能使用这些情况, 它们困扰着很多网页设计者。

本文旨在解决这些问题。文中首先讨论如何为 ActiveX 组件打包, 再论述如何为 ActiveX 组件进行运行时刻授权。

## 二、为 ActiveX 组件打包

在 WEB 网页内嵌入 ActiveX 组件时, 必然用到 < object > 标记。此标记的 CODEBASE 属性标明了如果用户的机器内还没有注册该组件, 那么浏览器应到哪个 WEB 服务器上下载该组件。

例如, 如查想为某个 ActiveX 组件设置其 CODEBASE 属性, 那么可以这样写:

```
CODEBASE= http://www.mysite.com/mycontrol.ocx
```

```
# version= 4 70 Q 1086
```

这种方式只从 WEB 服务器上下载组件的 .ocx 文件。由于组件通常还需要运行链接库 (DLL) 的支持, 因此用户的机器内如果没有相应的 DLL 或者没有相应版本的 DLL, 那么组件即使下载了, 也仍旧不能使用。

解决办法是采用 CAB 文件。此时, CODEBASE 属性可以这样写:

```
CODEBASE= http://www.mysite.com/mycontrol.cab  
# version= 1 2 0 0
```

CAB 文件通常包含组件的 .OCX 文件和一个 .INF 文件。INF 文件能够控制组件的安装和配置。当 .INF 文件发现用户的机器内没有组件运行所需要版本的 DLL 时, 它将自动到指定的 WEB 服务器上去下载这些 DLL 文件。因为 CAB 文件采用了压缩方法, 所以组件的下载时间比起直接下载 .ocx 文件要短。

如下是某个 CAB 文件内的 .INF 文件的内容 (不包含注释):

```
[version]           //CAB文件的版本信息  
signature= "$ CH ICA G O $ "  
AdvancedNF= 2 0  
[Add Code]         //CAB文件包含的组件和 DLL 信息  
ssdatb32.ocx= ssdatb32.ocx  
//该组件的 .ocx 文件存在于 CAB 文件内  
; mfc files  
//组件所需要的 DLL 文件, 它们不包含在 CAB 文件内  
msvert.dll= msvert.dll  
mfc42.dll= mfc42.dll  
olepro32.dll= olepro32.dll  
[ssdatb32.ocx] //组件的具体信息, 如注册 ID, 版本等
```

```
file-w in32-x86= thiscab
clsid= {BC496AED-9B4E-11CE-A6D5-0000C0BE9395}
FileVersion= 2,0,0,198
RegisterServer= yes
[msvcrt.dll]
```

组件所需 DLL 的版本

```
FileVersion= 4,20,0,6164
hook= mfc42installer
[mfc42.dll]
FileVersion= 4,2,0,6256
hook= mfc42installer
[olepro32.dll]
FileVersion= 4,2,0,6068
hook= mfc42installer
[mfc42installer]
```

/如果用户机器没有所需版本的 DLL,则到哪里下载这些 DLL文件

```
file-w in32-x86= http://activex.microsoft.com/controls/vc42/cab
```

在网页内如果要使用 ssdatb32.ocx时,其<object>标记的内容如下:

```
<OBJECT classid = clsid BC496AED-9B4E-11CE-A6D5-0000C0BE9395
codeBase= http://<% = Request.ServerVariables("SERVER_NAME")%>/ssdw3b32cab/height= 324
id= grid style= "HEIGHT: 250px; LEFT: 0px; TOP: 0px; WIDTH: 621px" width= 621>
<PARAM NAME= "ScriptBars"VALUE= "3">
<PARAM NAME= ".Version"VALUE= "196616">
<PARAM NAME= "DataMode"VALUE= "0">
<PARAM NAME= "Cols"VALUE= "0">
<PARAM NAME= "Rows"VALUE= "0">
.....
</object>
```

创建 CAB文件需要特定的工具,例如 Microsoft提供的 Cabinet Development Kit 该工具可以从 http://www.microsoft.com/workshop 站点上去下载。Cabinet Development Kit内有一个 cabarc.exe程序,此程序能够为 ActiveX 组件和 .NF文件打包,方法如下:

```
cabarc.exe N myocx.cab myocx.ocx myocx.inf
```

可以在一个 CAB文件内包含多个组件和 .NF文件。

利用 CAB文件能够使用户在浏览 WEB网页时顺利地找到和下载需要的组件极其所需要的 DLL,但组件能否正常运行还得看组件是否能够进行运行时刻授权。下节将讨论授权的问题

### 三、为 ActiveX组件进行运行时刻授权

大多数 ActiveX 组件支持设计时刻授权和运行时刻授权。设计时刻授权保证了程序员使用合法购买的组件设计程序;运行时刻授权保证了用户使用合法安装的组件运行程序。

当在开发环境中(如 VB VI等)为程序添加某个组件时将发生设计时刻授权。开发环境将检查待添加的组件是否是合法购买的(通常是寻找和分析一个 .lic的文件,该文件包含了组件的授权密钥(license key),然后把授权密钥同组件内部包含的密钥进行匹配比较)。对设计时刻授权来说,不管是 WEB程序开发工具,如 VI还是普遍应用程序开发工具,如 VB.VFP等,都使用完全相同的方式进行设计时刻授权。

运行时刻授权对于普通的应用程序和 WEB应用程序存在着很大的差异。

普通的应用程序,如 VB程序,其运行时刻授权可分为两个步骤。第一个步骤发生在为该 VB程序创建安装盘集时;第二个步骤发生在用户机器上安装好程序之后,程序被用户激活即将运行之前。在第一个步骤里,VB安装程序将把 ActiveX 组件的密钥复制一份到应用程序的二进制可执行文件中。由于是二进制文件,因此密钥被破解的可能性是很小的。在第二个步骤里,程序安装好后即将运行,此时,运行环境将从应用程序的可执行文件内分解出所要用的组件的密钥拷贝,然后把该密钥同相应组件内包含的密钥进行匹配比较(肯定是匹配的),这就保证了:一方面合法安装的程序能够可靠的运行和使用组件的功能,另一方面,用户不可能把组件用于程序设计(因为用户没有设计时刻授权所需要的 .lic文件)。

在 WEB应用程序中,如果在网页内嵌入了 ActiveX 组件对象,由于超文本是可阅读的,因此,WEB应用程序不可能把组件的密钥复制一份放到网页内的其它地方,程序员只能采用其它的方法。该方法是:在 WEB应用程序中使用 .LPK文件(license package file)。

一个典型的 LPK文件的内容如下:

```
LPK 许可协议包
//警告:本文件中的信息受著作权法和国际公约的 //
//保护。未经授权,随意复制或分发本文件, //
//或本文件的任何部分,都将承担严厉的法律风险。 //
//而且,您还不得对本文件进行反相工程。 //
//反编译或反汇编。 //
// //
{3d25aba1-caec-11cf-b34a-00aa00a28331}
oE1SW xQ 0hG 4skRFU IQAAA=
A gAAA=
gNw F rPF 90BGDngCgJK lL0hM AAAA
```

```

5ADUANAAXADQANAAXADcAQAAXAEEEAQABEAEUAN
-wAyAEMARABYAA=
4GpJvE6bz
hGm IQAAwL6TIRMAAAA5ADUANAAXADQANAAXAD
-cAQAAXAEEEAQABEAEUANwAyAEMARA
BYAA=

```

LPK 文件包含四个组成部分:

1. LPK 文件头, 标识 LPK 文件的类型
2. LPK 文件的授权和警告信息, 用作为书面形式上的反拷贝宣传。
3. LPK 版本 GUID, GUID 一方面标识 LPK 数据信息的开始, 另一方面标识 LPK 文件的版本格式。
4. LPK 文件的数据信息, 这些信息是经过加密处理的, 它们包含了 WEB 网页中使用的组件的密钥。

IE4.0 及其以后的版本自带一个叫做 ActiveX License Manager 的组件。该组件能够从 LPK 文件中分解出组件的密钥, 然后把此密钥同网页内相应组件包含的密钥做匹配比较, 从而保证了浏览器能够正确使用和运行组件的功能。

使用 LPK 文件的方法是在含有 ActiveX 组件的网页内嵌入一个 ActiveX License Manager 组件 (该组件不用考虑授权问题, 因为 IE 本身含有此组件), 并设置它的 LPKPath 属性为需要的 LPK 文件。假设已经有了组件的 LPK 文件, 则可以在网页内加入:

```

<OBJECT CLASSID= "clsid: 5220cb21-c88d-11cf-b347-00aa00a28331">
<PARAM NAME= "LPK Path" VALUE= "relative URL to LPK file">
</OBJECT>

```

其中 CLASSID 部分是 ActiveX License Manager 的系统注册号。

采用 LPK 文件对 WEB 网页内的 ActiveX 组件进行运行时刻授权有两个方面必须注意:

1. 每个网页内只能使用一个 LPK 文件, 但每个 LPK 文件可以包含多个组件的授权信息;

2. 在网页内 ActiveX License Manager 组件应该放在所有需要授权的组件之前。例如, 在笔者的网页中使用了 ss-datb32.ocx 组件, 且已经为该组件创建了一个 LPK 文件为 GRID.LPK, 该 LPK 文件与网页放在同一个目录下, 则在网页中可以写入:

```

<OBJECT classid = clsid: 5220cb21-c88d-11cf-b347-00aa00a28331>
<PARAM NAME= "LPK Path" VALUE= "grid.LPK">
</OBJECT>
<OBJECT align= center
classid = clsid: C589DD00-0F28-11D1-B230-0000C08C00C4

```

```

codeBase= http://<%= Request.ServerVariables
("SERVER_NAME")%>/ssdw3
b32cabheight= 324 id= grid
style= "HEIGHT: 250px; LEFT: 0px; TOP: 0px;
WIDTH: 621px" width= 621>
<PARAM NAME= "ScrollBars" VALUE= "3">
<PARAM NAME= "Version" VALUE= "196616">
<PARAM NAME= "DataMode" VALUE= "0">
.....
</OBJECT>

```

利用微软的 ActiveX SDK (可从 http://www.microsoft.com/activex 上下载) 中的 LPK-TOOL.EXE 程序可以创建 LPK 文件。

运行 LPD-TOOL.EXE 将弹出一个对话框, 如图 1 对话框内左边的组合框列出系统中所有能够进行运行时刻授权的 ActiveX 组件, 只需用 ADD 或者 REMOVE 按钮就可以轻松地把需要运行时刻授权的组件放入左边的授权组件列表框内。单击 Save&Edit 按钮将生成 LPK 文件, 而且 LPK 文件的文件名是可以随意指定的, LPK 文件创建好以后, 就可以根据需要按照文中讨论的方法加入到 WEB 网页中。

### 四、总结

正确地使用 CAB 文件和 LPK 文件, 可以彻底解决 ActiveX 组件的下载和运行时刻授权问题, 并且用户不可能把组件非法用在自己的程序设计之中。有了文中的详细讨论, 相信读者能够灵活自如的在 WEB 网页内使用各种 ActiveX 组件。

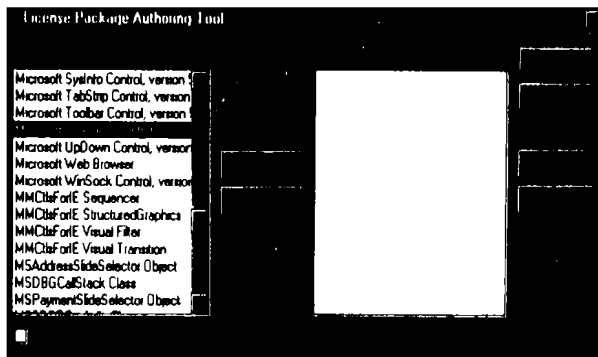


图 1 LPKTOOL 对话框

### 参考文献

[1] 美 Microsoft 公司, << Microsoft Developer Network >>, Microsoft Press, 1994-1997

(收稿日期: 99年 2月 5日)