

VJ++ 中 CAB 文件的创建及使用

张世辉 胡笔蕊 李永强 (燕山大学计算机系 秦皇岛 066004)

摘要:本文介绍了 VJ++ 中 CAB 文件创建及使用的方法,解决了使用 COM 组件的 Java Applet 仅能在 VJ++ 集成开发环境下运行的问题。

关键词:DDF 文件 CAB 文件 签名

1. 引言

使用 COM (Component Object Model) 组件的 Java Applet 如果在 VJ++ 集成开发环境下运行,那么所有的类文件都是可信的,这对 Applet 开发期间的调试是很有益处的。如果想把使用 COM 组件的 Java Applet 提供给 Internet/Intranet 上的其他 Web 用户,由于涉及到安全性的问题,必须把它放在一个经过签名的 CAB 文件中,否则将不能正常执行。结合开发实例,我们有一个使用了 COM 组件的 Java 项目 access,其 HTML 文档 access.html 的源代码如下:

```
//access.html
<html>
<head>
<title>使用 COM 组件的 Java 项目</title>
</head>
<body>
<center>
<applet
  code = access.class
  id = access
  width = 600
  height = 250>
</applet>
</center>
</body>
</html>
```

该项目涉及到的类文件有 access.class、layout.class、mypaint.class、databasefield.class。在集成开发环境下,该项目可正常执行,若脱离此环境,该 Java 项目将不能正常执行,因为 Java 虚拟机仅允许可信任的类文件使用 COM 组件,经过数字签名的 CAB 文件是可信的。下面结合该项目来阐述 CAB 文件的创建及使用方法。

2. 用所需类文件创建 CAB 文件

为了装配 CAB 文件,首先需要创建一个 DDF 文件。DDF 文件是一个文本文件,它描述了将要创建的 CAB 文件的布局。DDF 文件可繁可简,一般如下创建即可满足需要。

```
//DDF 文件的内容
.OPTION EXPLICIT
.Set CabinetNameTemplate = CABNAME.cab
.Set DiskDirectoryTemplate =
.Set Cabinet = on
.Set Compress = on
.Set ReservePerCabinetSize = 6144
CLASSNAME.class
CLASSNAME.class
```

此处 CABNAME 是将要创建的 CAB 文件名字, CLASSNAME.class 项形成了一个包含于 CAB 文件的类文件列表。结合实例,我们的 DDF 文件具体如下:

```
//文件名:mycab.ddf
.OPTION EXPLICIT
.Set CabinetNameTemplate = mycab.cab
.Set DiskDirectoryTemplate =
.Set Cabinet = on
.Set Compress = on
.Set ReservePerCabinetSize = 6144
access.class
layout.class
mypaint.class
databasefield.class
```

创建完 DDF 文件后,在当前项目目录下,运行下面命令就可创建 CAB 文件“mycab.cab”。

```
c:\cabdir\diamond /f mycab.ddf
```

这里假设已经安装了文件柜开发工具箱于目录“c:\cabdir”中,如果还未安装该工具箱,可在 VJ++ 光盘的目录“\CAB&SIGN”中找到。

显而易见,CAB 文件使用起来是很方便的。由于

CAB 文件中的类文件是以压缩格式存储的,故下载过程比较快。另外,多个类文件存放在一个 CAB 文件中也提高了执行效率,因为这样做减少了网络存取操作的次数。

3. 创建测试证书

在对 CAB 文件进行签名之前,需要创建公司测试证书,公司测试证书通常包括公司的一般信息如联系信息等。运行下列命令可以创建测试证书。

```
C: \ sign \ MAKECERT - u: key - n: CN = SoftwareCompany - k: key. pvk cert. cer
```

```
C: \ sign \ CERT2SPC c: \ sign \ root. cer cert. cer cert. spc
```

需要注意的是,“MAKECERT”命令对大小写是敏感的,必须使用小写的 -u、-n 和 -k,并且必须是大写的 CN=。另外,这里假设已经安装了代码签名工具箱于目录“c: \ sign”中,如果还未安装该工具箱,可在 VJ + + 光盘的目录“\ CAB&SIGN”中找到。

命令“MAKECERT”创建了测试证书文件“cert. cer”和相关的私人钥匙文件“key. pvk”,程序开发者可修改这两个文件名为自己喜欢的名字。私人钥匙文件“key. pvk”中存有该证书的“key”值。如果曾对文件名“cert. cer”做过修改,那么命令“CERT2SPC”中的参数也应做相应的调整。

命令“CERT2SPC”由测试证书文件“cert. cer”创建了软件发行证书文件“cert. spc”,在此过程中从文件“root. cer”中获得测试证书管理机构的有关信息。

稍后就是利用私人钥匙文件“key. pvk”和软件发行证书文件“cert. spc”对 CAB 文件进行签名的。

结合实例,我们的“MAKECERT”和“CERT2SPC”命令如下:

```
C: \ sign \ MAKECERT - u: 19731204 - n: CN = sigCO. - k: mykey. pvk mycert. cer
```

```
C: \ sign \ CERT2SPC c: \ sign \ root. cer mycert. cer mycert. spc
```

4. 对 CAB 文件进行签名

一旦拥有了测试证书,对 CAB 文件进行签名将是一件很容易的事情,可以利用代码签名向导“signcode”来实现签名。下面结合实例进行讲解,步骤如下:

① 如下运行“signcode”,出现代码签名向导对话框;

```
C: \ sign \ signcode
```

② 单击“Next”按钮;

③ 在第一个编辑框中,输入想要签名的 CAB 文件名,结合实例应为“mycab. cab”,可用“浏览”功能定位该

文件;

④ 在第二个编辑框中,为此 CAB 文件输入一个名字,这个名字将出现在证书上;

⑤ 单击“Next”按钮;

⑥ 在第一个编辑框中输入 SPC 文件的名称,结合实例应为“mycert. spc”,可用“浏览”功能定位该文件;

⑦ 在第二个编辑框中输入 PVK 文件的名称,结合实例应为“mykey. pvk”,可用“浏览”功能定位该文件;

⑧ 在第二个编辑框的下面任选一种在签名过程中使用的加密算法;

⑨ 连续两次单击“Next”按钮;

⑩ 单击“Sign”按钮对 CAB 文件进行签名。

这样,DDF 文件中的所有类文件均已放入 CAB 文件中且进行了数字签名。需要注意的是,在上述③、⑥、⑦中列出的文件名都是最终的文件名,实际应用中最好将该文件所在的目录及网络地址也包括进去,以便于其他 Web 用户共享。

5. CAB 文件的使用

为了使用签名后可信的 CAB 文件中的类文件,可以在 HTML 文档的 <applet> 标签中加入一个参数标明该 CAB 文件即可,结合实例,修改后的 access. html 文档如下:

```
//修改后的 access. html
```

```
<html>
<head>
<title>使用 CAB 文件的 Java 项目</title>
</head>
<body>
<center>
<applet
code = access. class
id = access
width = 600
height = 250>
<param name = "cabbase" value = "mycab. cab">
</applet>
</center>
</body>
</html>
```

这样,使用 COM 组件的该 Java 项目不仅可在 VJ + + 集成开发环境下正常执行,也可供网上其他 Web 用户使用了。故利用 CAB 文件,可以解决使用 COM 组件的 Java Applet 仅能在 VJ + + 集成开发环境下运行的问题。

(来稿时间:1998年10月)