

优化 ActiveX 控件

抚顺石油学院计算机系 汤宏鹏 王 杪

一、为了 Internet Explorer 4.0 而优化 ActiveX 控件

本文的核心内容是如何改进 ActiveX 控件,使它们能与最新版本的 Internet Explorer 充分合作。

为 Internet Explorer 4.0 优化 ActiveX 控件—规则已经变了,但不是很多 Internet Explorer 4.0 带来一些新的编写控件的规则,因此运行得相当理想。旧的控件仍将工作,但是您会发现,一些更改过的内容可以帮助您的控件和用户系统运行得更好。其中的两个更改特别重要:至少使用了线程化套件模式以及支持无窗口式操作。

为什么要为了 Internet Explorer 4.0 而费力优化控件呢?有两个原因:首先,Internet Explorer 4.0 已成为最流行的浏览器。其次,Internet Explorer 4.0 Dynamic HTML 控件将应用于 Active Desktop 和多种应用程序中,包括 e-mail、HTML Help、Visual Studio IDE 和很多第三方应用程序。

二、未优化控件的缺陷

控件是否显示出需要 Internet Explorer 4.0 优化的症状?下面是必须优化的缺陷:

当 Internet Explorer 运行控件的多个实例时,系统性能降低(也称为“套件嫉妒”);

完全无视 Internet Explorer 4.0 的多媒体过滤和转换效果(也称为“窗口封闭”);

使用虚假的窗口句柄(也称为“窗口激活混乱”);

缺乏持续和敏捷性(也称为“属性混乱”);

缺乏同 Internet Explorer 4.0 周围的文档对象模型的交互性(也称为“DOM 不足”);

在安全环境中功能失败(也称为“签名/标记不足”);

提交和安装缓慢或缺乏(也称为“过缓提交”);

中断当前的应用程序(也称为“激活和焦点偷盗狂”);

使人无法知道它的需要(也称为“事件启动混乱”);

如果控件显现出一种或几种这样的症状,那么就必须进行优化。幸运的是这些缺陷不是致命的,并且使用下面的一种或几种技术就能较容易地改正。

三、编写控件的方法

编写 ActiveX 控件有三种主要方法:使用 C++ 和 MFC、C++ 和 ATL 或者 Visual Basic。您可能不知道的是在 Internet Explorer 4.0 中,任何 Java Bean 都自动成为一个 ActiveX 控件,因此您还可以使用 Java——即使用于可视控件也可以(其他工具,例如 Delphi,也支持 ActiveX 控件的编写。).

要想得到最好最快的控件,最好的方法是:使用 ATL 2.0,即 Active Template Library,它与 Visual C++ 5.0 一起发行。对于简单并且相对较小的控件,可以使用 MFC(但是您的控件需要在用户的计算机上安装 MFC DLL)。Visual Basic 易于编写,但是需要一个大型的 DLL,并且不能提供 C++ 技术的漂亮控件。Java 控件需要 Java VM(包含在 Internet Explorer 中),并且有些慢,也不能提供漂亮的控件。

我们将集中讨论 ATL 和 MFC 控件。

1. 使用多线程化套件模式

这是一个大问题,因为忽略了这个问题就必然会降低用户操作的性能。所以,确保您的控件使用多线程化套件模式很重要。很多旧的控件是单线程的,当这种控件的多个实例通过 Internet Explorer 4.0 Web 浏览器(这是目前最流行的)控件的多个实例而运行时,会引起严重的性能问题(对于 Internet Explorer 3.0 这也是一个问题,但不明显,因为不经常运行该浏览器的多个副本。).

默认情况下,使用 MFC(4.2 版本或更新版本)和最新更新的 Visual Basic 4.0 创建的控件,会自动具有多线程化套件模式特性。如果您正在使用 ATL,那真是很幸运:您可以在创建控件时进行选择。如果您使用的是旧版本的 MFC 或 Visual Basic 生成的控件,升级到最新版本并重新生成这些控件就很容易——这是您更新控件时所需的大部分工作。不过别忘了在注册表中将您的控件标记为具有多线程化套件模式特性。

但是,必须注意另一件重要的事情:由于 MFC、ATL 和 Visual Basic 在运行时不允许多线程访问所引起的重进入现象,所以必须保护全局和静态数据。为此,我的处方是:使用合适的同步技术(象旗语信号一类的)来保护所有的全局和静态数据,因为这样能保证某一时刻只有一个线程手术刀在进行操作。

2. 使控件无窗口

对 MFC 和 ATL 控件需要做的下一个改变是:使它们无窗口——这样它们会在容器(而非自己)的设备中绘制自己。不幸的是在当前版本中,Visual Basic 控件不能是无窗口式的。因此要创建一个无窗口式控件,您必须使用 MFC 或 ATL。

除了无窗口式控件运行更快以外,控件无窗口还有几个好处。对于 Internet Explorer 4.0 来说,最重要的是无窗口式控件可以应用多媒体过滤器和转换效果。如果控件有自己的窗口,Internet Explorer 4.0 就不能对该控件应用这种

很酷的多媒体效果——这可就束缚了使用这些控件的人的手脚。

除了这些,无窗口式控件可以是非矩形的和透明的(并且 Internet Explorer 4.0 可以控制它们的前后顺序(哪个控件在最前面))。

3. 制作自己的窗口

Internet Explorer 4.0 也改变了控件的激活方式:只有可见时才能是活动的。这意味着,如果需要在窗口中显示信息,就需要创建自己的窗口;除非可见,否则您不能得到普通的活动窗口。

4. 准备好后让 Internet Explorer 知道

在 Internet Explorer 4.0 中,有一种标准的方法可以给出有关控件何时准备好的详细信息:只需实施 DISPID - READYSTATE 自动服务属性,并在准备好改变状态时调用容器的 IPropertyNotifySink::OnChanged。注意,如果已进入准备就绪状态,就必须保证能正确通知容器。如果不这样做,将永远不会调用 window.onload 脚本方法程序。只有在所有实施准备就绪状态的对象,都表明它们都已经准备好时,才能调用这种方法程序。

5. 使用 IPersistHistory

在 Internet Explorer 3.0 中,有一个四个网页的“页缓存”。这意味着 ActiveX 控件(和 Java 小程序)会在用户漫游到另三个网页时仍是活动的。这很便捷,因为这允许用户漫游回来,并且保持控件的状态同离开时一样。例如,如果控件包含用户正在编辑的内容,当用户回来时控件的内容将保持不变。

但是,这种缓冲会引起性能问题,因此在 Internet Explorer 4.0 中被抛弃了。而这意味着控件会得健忘症——它们丧失记忆——即使您只离开了一个网页。但是有一个处方,只需举手之劳即可解决问题,并且比以前的效果更好。

当控件启动时,调用 IPersistHistory::LoadHistory 以确定浏览器控件是否已经加载了数据。如果调用成功了,它会返回一个 IStream 接口,从中可以进行阅读。

当由于浏览器移向其他网页而使您的控件将被破坏时,可调用 IPersistHistory::SaveHistory 以得到 IStream 接口,您可以向这上面写下需要保存的数据。

IPersistHistory 记录的数据只用于同一浏览器工作期,而不管用户漫游了多少网页(这比任意限制网页数目要好得多。)。但是不要用这种机制写下太多的数据:浏览器所有实例中所有网页的系统限制总共只有 2 MB。对于大量数据,例如图像和数据库查询结果,可使用缓存 API 将数据文件放在缓存中,并且只用 IPersistHistory 写下对象的标识符。

6. 利用 Dynamic HTML

控件很容易访问 Web 页上的 Dynamic HTML 对象:调用 IOleClientSite::GetContainer 以访问容器对象,然后使用 QueryInterface 以获取指向 IHTMLDocument2 界面的指针。这样就可以访问甚至修改 Web 页上的每一个对象。将源代码展露为脚本将是一种吸引人的变通方法。注意,在脚本中很容易设计原型和调试代码,然后将脚本翻译成界面调用。

7. 工作签名

即使控件不安全,为控件签名也是很重要的(如果它们不够安全,不要将它们标记为安全的!)。签名和标记是两种不同的事情:签名只说明编写了一个控件,标记它就进一步声明该控件在任何 Web 页上都可以安全使用。可以将编写控件看作是画了一幅好画:哪个艺术家不为自己的作品签名呢?浏览器可以设置为简单地忽略未签名的控件,所以签上您的名字就更重要了(实际上,高级和中级安全设置都完全忽略未签名的控件。)

在将您的控件标记为安全的之前(通过实施 IObject Safety),要确定它是真正安全的。安全性有两种:初始化的安全和脚本的安全。初始化的安全意味着能引起控件破坏用户计算机的初始化输入流不可能存在(例如,能让一个输入值引起数组超出它的维数吗?)。脚本的安全意味着不可能编写能引起控件破坏用户计算机的脚本命令和脚本命令组合(需特别注意无效的参数值。)

虽然下面的列表不完全,但是这些行为几乎总是不安全的:

读取,特别是写入磁盘或注册表

在内存分配之外读取,特别是写入

跨帧(Cross-frame)访问任何内容

如果不能将控件标记为安全的,也要签名,这样用户可以在适当的安全区域使用它。

8. 分发控件

向用户的计算机分发控件和其他程序最好的方法是使用开放式软件描述(Open Software Description)标准。OSD 是一种 XML 用语,由 Marimba 和 Microsoft 设计,Netscape、Lotus、BackWeb 等都支持它。您可以将 OSD 内容作为单独的文件或者作为一组内容定义格式(CDF)标记的一部分来分发。OSD 将支持分发的推(push)或拉(pull)技术。

对于 Internet Explorer 3.0,您需要创作 <OBJECT> 标记,使它指向一个其中含有嵌入的 .INF 文件的 .CAB 文件。对于 Internet Explorer 4.0,您可以在 CAB 中嵌入 OSD 及 INF 文件,或者引用一个 CDF 文件。注意,这种新方法也自动提供了卸载功能。

不依赖于用户的计算机上是否装有 DLL(如 MFC、运行时时刻的 C、运行时时刻的 Visual Basic 或者运行时时刻的 ATL),这一点很重要。即使用户的计算机上没有所需的

DLL, 向 INF 文件中加入几行代码, 让正确的 CAB 文件从 Microsoft 下载也是很简单的。这样, 只有那些还没有这种 DLL 的用户才必须花时间得到它。

9. 不要从活动的应用程序中盗窃 activation 或焦点

越来越多的极其粗暴的程序在您正在做其他事情时, 突然在您面前弹出一个对话框。更糟糕的是, 这些程序甚至盗窃键盘焦点, 这意味着您向应用程序中正在键入的字符, 最后是由这个对话框处理了。

在一个任何网页都可以作为桌面的环境里, 控件实际上是放在用户的桌面上。所以从不异步地弹出对话框就显得非常重要。只在响应用户操作时才打开对话框。这确实是良好的设计, 但对于桌面上的控件来说这非常重要。

不要打开一个对话框, 而是改变控件甚至任务栏的视觉状态。例如, 如果对于您的股票行情接收控件, 服务器还没有准备好, 只需在控件中显示一条信息说“服务器不可用”, 而不要用一个不必要的、讨厌的对话框打断用户的操作。如果情况很重要, 您可以发出声音来警告用户。不要只是弹出一个对话框, 从用户正在使用的应用程序中盗窃 activation 或焦点。

10. 其他普遍的问题

有很多普遍问题, 这些问题是在 Internet Explorer 4.0 中使用 3.0 版的控件时出现的。请格外留意这些普遍问题: 记住, 控件只有在可见时才能被激活。

避免使用 XObject 所用的名称。

只从创建了控件的线程中调用容器 (引发事件)。如果

控件是套件模式的, 并且未启动自己的任何线程, 这就不是问题, 因为只能在自己的线程中被调用。但是如没有线程, 或者启动了自己的线程, 就请注意, 如果在不同的线程中引发了一个事件, 容器将永远也看不到该事件。当容器请求 FreezeEvents 时, 不要引发任何事件。如果引发了, 这些事件会被忽略 (就象在交谈时一样。)。注意, 您必须跟踪自己被要求冻结事件的频率, 并且只在计数为零时才引发事件。

使用 Active Desktop 进行调试

启动 Active Desktop 时调试控件是一种特殊的挑战, 因为 IDE 调试器不加载浏览器——它已经加载了。这使得在控件中设置断点变得很困难。最简单的选择是在调试时禁用 Active Desktop。如果不愿这样做, 您可以试着加入 Active Desktop 过程, 或者使用“任务管理器”关闭 Active Desktop (Explorer 外壳) 过程。如果您关闭了 Active Desktop, 有很多方法可以重新启动它, 包括双击桌面, 按下 Ctrl+Shift+Esc, 以及使用“任务管理器”启动。

参考文献

[1] David Chappell, Understanding ActiveX(tm) and OLE Microsoft Press, Release 9/17/96 ISBN# 1-57231-216-5

[2] Kirtland, Mary, Designing Component-Based Applications. Microsoft Press, 1998. ISBN 0-7356-0523-8.