**Связка ActiveX - Internet Explorer**

Иван Семенов

А знаете ли вы, что на Delphi можно писать ActiveX компоненты? Конечно знаете. А что с их помощью можно взаимодействовать с Internet Explorer? Это может быть интересно для профессиональных вебмастеров, скажете вы, но я не согласен. "Простой" программист тоже может найти массу применений этому. Здесь будет описано одно из них. Все мы лазим (ходим и т.д.) по интернету. И вы тоже - раз читаете эти строки :). А не случалось ли вам, случайно где-то побывав, что-то прочитав и благополучно забыв адрес сайта через некоторое время вдруг понять, что там было именно то, что вам сейчас срочно понадобилось? Можно конечно посмотреть History браузера, можно залезть в кэш "руками" и попытаться найти там что-то. А можно написать компонент, который бы искал слова в файлах кэша (в общем случае в любых HTML-файлах) и выводил бы на просмотр требуемые файлы. Связать этот компонент с Эксплорером - и вперед. Что удобно - вся работа происходит в эксплорере: и поиск, и,естественно, просмотр. При этом для Delphi-программиста не нужны особые знания языка HTML, скриптовых языков и т.п. Достаточно знать несколько основных конструкций (а уж справочных руководств в интернете очень много). Написанный ActiveX-компонент вставляется в HTML-страничку. Вот пример простейшей странички

<HTML>

<HEAD>

<TITLE>Поиск</TITLE>

</HEAD>

<BODY>

<P ALIGN=CENTER>

<OBJECT ID="findword1" - {при помощи этого тэга компонент вставляется в страничку}

CLASSID="CLSID:47E50425-E611-11D3-970A-4854E82B17E6"

CODEBASE="C:\PATH\FINDWORDS.OCX">

</OBJECT>

</P>

</BODY>

</HTML>

В этом примере ActiveX-компонент, находящийся в файле C:\PATH\FINDWORDS.OCX вставляется в HTML-страничку. Но важно отметить, что эта страничка откроется только в Microsoft Internet Explorer версии 4 и старше. Пишут, что третий эксплорер тоже поддерживает тэг <OBJECT>, но сам не пробовал, не знаю. Браузеры Netscape, Opera и какие еще там бывают, его не поддерживают.

Итак, тэг <OBJECT> вставляет в страничку ActiveX-компонент. Его атрибут CLASSID указывает идентификатор класса нашего компонента. При создании в Delphi компонента с нуля ему автоматически присваивается этот идентификатор класса. ID="findword1" - имя объекта. Здесь можно писать любое имя. По нему мы в дальнейшем будем ссылаться на наш компонент в теле странички из скриптов-процедур обработки событий. Далее, для того, чтобы наш компонент мог использоваться прикладными программами, он должен быть зарегистрирован в реестре. Зарегистрировать его можно программой regsvr32, которая по умолчанию находится в каталоге [System]. Например так: [regsvr32 C:\PATH\FINDWORDS.OCX]. Если при открытии странички Explorer не находит в реестре указанный компонент, то он ищет его в местоположении, указанном атрибутом CODEBASE. Здесь может быть полный путь к файлу, если он находится на вашем жестком диске или даже URL-адрес (со всеми сопутствующими атрибутами, как то http:// и т.д.).Т.е, если эксплорер встретил ссылку на компонент, а этого компонента нет на вашей машине, он может загрузить его из интернета с указанного адреса. Кстати, атрибут CLASSID - обязательный, именно по нему производится "идентификация" класса. А атрибут CODEBASE - необязательный. В случае, когда он опущен, если компонент уже зарегистрирован в системе, то он отобразится в вашей страничке, если не зарегистрирован - страничка будет пустой. И наконец если эксплорер сам регистрирует компонент, он переписывает файл OCX в папку [Windows\Downloaded program files].

Для того, чтобы вручную не писать скрипты подсоединения ActiveX компонентов, я советую скачать программу Microsoft ActiveX Control Pad отсюда. Эта программа предназначена для внедрения ActiveX-компонентов в HTML-странички. После ее работы определение компонента выглядит примерно так:

<OBJECT ID="findword1"

CLASSID="CLSID:47E50425-E611-11D3-970A-4854E82B17E6"

CODEBASE="C:\PATH\FINDWORDS.OCX">

<PARAM NAME="Visible" VALUE="-1">

<PARAM NAME="AutoScroll" VALUE="0">

<PARAM NAME="AutoSize" VALUE="0">

<PARAM NAME="AxBorderStyle" VALUE="1">

<PARAM NAME="Caption" VALUE="findword">

<PARAM NAME="Color" VALUE="2147483663">

<PARAM NAME="Font" VALUE="MS Sans Serif">

<PARAM NAME="KeyPreview" VALUE="0">

<PARAM NAME="PixelsPerInch" VALUE="96">

<PARAM NAME="PrintScale" VALUE="1">

<PARAM NAME="Scaled" VALUE="-1">

<PARAM NAME="DropTarget" VALUE="0">

<PARAM NAME="HelpFile" VALUE="">

<PARAM NAME="DoubleBuffered" VALUE="0">

<PARAM NAME="Enabled" VALUE="-1">

<PARAM NAME="BiDiMode" VALUE="0">

<PARAM NAME="Cursor" VALUE="0">

<PARAM NAME="filename" VALUE="nothing">

</OBJECT>

Т.е. эта программа сама подставляет полное определение компонента (его CLASSID, например). Правда, полученный код иногда приходится подправлять вручную. Например может потребоваться убрать явное указание высоты и ширины объекта.

Теперь подходим к самому главному: как сделать сам компонент (чтобы было что вставлять в нашу страничку :). Итак, в Delphi делаем New\ActiveX\Active form. В окошке Active Form Wizard выбираем Threading model=Apartment. Другие threading models не работают с IE 4. Выглядит это так: компонент в страничке открывается, но иногда вдруг выскакивает Access violation. (обычно на событие Create). Модель же Both работает с IE 5. Флажок "Include Design-Time licence" лучше не устанавливать. Дальше открывается новая форма, где вы можете размещать свои кнопки-текстбоксы, определять реакцию на события и т.д.

Далее будут описаны некоторые хитрости. Например, нужно хранить некоторые данные во внешнем файле. Я столкнулся со следующим: мой компонент на разных машинах размещал свои файлы в разных местах: на одной в каталоге Windows, на другой - на рабочем столе. Был найден такой выход: пусть страничка по требованию компонента возвращает ему каталог, в котором она находится. Для этого на форму я поместил PageControl, сделал закладки невидимыми и на OnShow (у формы ActiveX компонента нет события OnShow) одной из страниц поставил генерацию собственного события OnWantDir. А в теле HTML-странички соответственно реакцию на него:

<SCRIPT LANGUAGE="VBScript">

<!--

Sub findword1\_OnWantDir()

findword1.page\_location = location.href

end sub

-->

</SCRIPT>

Далее, это событие OnShow происходит сразу после создания экземпляра компонента. Так вот, если событие OnWantDir генерировать непосредственно в нем (в OnShow), то видимо что-то в недрах Windows не успевает провернуться и машина виснет. Поэтому пришлось повесить на форму таймер, на OnShow таймер запускать, и уже на OnTimer как раз и вызывать свое событие OnWantDir. Интервал у таймера я поставил в полсекунды. Конечно можно было бы хранить свои файлы например в каталоге [Windows], но почему-то функция GetWindowsDirectory при вызове из ActiveX-компонента возвращала ошибку, хотя тут же нормально отрабатывала из обыкновенного приложения (exe). То же и с GetSystemDirectory и GetTempDirectory.

Как сделать компонент тиражируемым? Чтобы пользователь смог работать с ним сразу же, не запуская никаких дополнительных программ, не указывая всяких-разных путей и т.д. Вот пример HTML-странички (а здесь его скриншот):

<html>

<HEAD>

<title>Поиск</title>

<SCRIPT LANGUAGE="VBScript">

<!--

Sub Procedure1()

location.href = findword1.NewStroke

{Получить от компонента имя файла и открыть его для просмотра. Эта процедура запускается при возникновении события OnDocClick. Location - объект Explorer'а (см. документацию по VBScript)}

end sub

-->

</SCRIPT>

</HEAD>

<SCRIPT LANGUAGE="VBScript">

<!--

Sub findword1\_OnWantDir()

findword1.page\_location = location.href

{Получить текущий каталог, т.е. свойству page\_location объекта присвоить местоположение нашей странички}

end sub

Sub findword1\_OnDocClick()

{При возникновении события OnDocClick вызвать процедуру Procedure1 (открыть файл для просмотра)}

call Procedure1()

end sub

-->

</SCRIPT>

<p align = "center">

<OBJECT ID="findword1"

CLASSID="CLSID:47E50425-E611-11D3-970A-4854E82B17E6"

CODEBASE="findwords.ocx">

{Здесь просто имя файла без пути. Explorer зарегистрирует компонент невидимо для пользователя, взяв его из текущеего каталога (страничка и файл OCX находятся в одном каталоге)}

<PARAM NAME="Visible" VALUE="-1">

<PARAM NAME="AutoScroll" VALUE="0">

<PARAM NAME="AutoSize" VALUE="0">

<PARAM NAME="AxBorderStyle" VALUE="1">

<PARAM NAME="Caption" VALUE="findword">

<PARAM NAME="Color" VALUE="2147483663">

<PARAM NAME="Font" VALUE="MS Sans Serif">

<PARAM NAME="KeyPreview" VALUE="0">

<PARAM NAME="PixelsPerInch" VALUE="96">

<PARAM NAME="PrintScale" VALUE="1">

<PARAM NAME="Scaled" VALUE="-1">

<PARAM NAME="DropTarget" VALUE="0">

<PARAM NAME="DoubleBuffered" VALUE="0">

<PARAM NAME="Enabled" VALUE="-1">

<PARAM NAME="BiDiMode" VALUE="0">

<PARAM NAME="Cursor" VALUE="0">

<PARAM NAME="filename" VALUE="nothing">

<PARAM NAME="page\_location" VALUE="">

</OBJECT>

</p>

</BODY>

</html>

И еще раз: 1) открываем нашу страничку (в IE 4 и выше); 2) если компонент зарегистрирован, он сразу показывается, если не зарегистрирован, то регистрируется и показывается. При этом: 3) после создания выдерживается пауза в полсекунды и запрашивается текущий каталог (и страничка и сам OCX-файл находятся в одном каталоге, который и будет текущим). 4) если нужно открыть на просмотр какую либо страничку (выбранную пользователем в процессе работы из списка - см. скриншот), то свойству компонента (при внедрении его в страничку правильнее будет называть его уже объектом) присваивается значение (имя файла), генерируется событие. Cкрипт-обработчик этого события читает свойство и отрывает требуемый файл.