15 Seconds : Integrating Flash with an Access Database
Matt Duckhouse
06/05 / 2001

Macromedia's Flash is often only used to create those annoying Web site intros that people skip right past. Well, since the latest versions of Flash including the ability to interface with ASP and other server-generated Web pages you can now do much more.

This article explains how to connect a Flash movie to an Access database, and use an ASP page to query the database and hand information over to the Flash movie. We are going to build a very simple Flash address book to demonstrate the technique.

You are going to need a few tools to build the address book: Macromedia Flash 5, Internet Information Services 4.0 (or IIS 5.0), and a copy of Microsoft Access.

You can both download all the source files and see the final product here.

The Basics

A Flash movie cannot query a database directly. It can, however, fetch an ASP page that, in turn, can query a database. This functionality revolves around using Flash's ActionScript function loadVariables, as follows:

`loadVariables(URL, location);`

The loadVariables function retrieves the contents of the URL specified and used this to set variables within the Flash movie. The content must be in Multipurpose Internet Mail Extension (MIME) format or (to get technical) application/x-www-urlformencoded. For instance, if the URL specified contains a page with the following content:

`Var1=Test&Var2=Demo`

The variable Var1 within the Flash movie would be set to "Test" and the variable Var2 would be set to "Demo." The variables can then be accessed through Flash ActionScript to modify the behavior of the movie. In our demonstration we are going to use this behavior to pass data to a Flash movie from an Access database that will be queried by an ASP page.

Database Design

First, let's build the database. Our address book is going to be pretty simple so the database only has a single table called "Contacts" with five fields: ContactID, Name, Telephone, City, and Notes.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContactID</td>
<td>AutoNumber</td>
<td></td>
</tr>
</tbody>
</table>
The database is called "AddressBook.mdb" and is held in the same directory as the ASP and Macromedia Flash File Format (SWF) files we are about to build. (SWF is the file format used by Macromedia Flash to deliver graphics, animation, and sound over the Internet. About 90 percent of Web users can view SWF content without having to install a plug-in.)

**ASP Design**

Let's take a look at the ASP page we will use to fetch

```<%
Set DataConn = Server.CreateObject("ADODB.Connection")
DataConn.Open "Driver=Microsoft Access Driver (*.mdb);DBQ=" & Server.MapPath("AddressBook.mdb")

Set cmdTemp = Server.CreateObject("ADODB.Command")
Set rstContacts = Server.CreateObject("ADODB.Recordset")

cmdTemp.CommandText = "Select * From Contacts"
cmdTemp.CommandType = 1
Set cmdTemp.ActiveConnection = DataConn

rstContacts.Open cmdTemp, , 1, 3
rstContacts.Move CLng(Request("Record"))
Response.write "Name=" & Server.URLEncode(rstContacts("Name")) & 
Response.write "Telephone=" & Server.URLEncode(rstContacts("Telephone")) & 
Response.write "City=" & Server.URLEncode(rstContacts("City")) & 
Response.write "Notes=" & Server.URLEncode(rstContacts("Notes")) & 
Response.write "TotalRecords=" & rstContacts.RecordCount

rstContacts.Close
DataConn.Close
%>```

The page assumes that we pass in the record that we want back from the database and then it returns the information in MIME format using the Server.URLEncode command.

A typical output from our ASP page might be:

<table>
<thead>
<tr>
<th>Name</th>
<th>Text</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telephone</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>City</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Notes</td>
<td>Memo</td>
<td>-</td>
</tr>
</tbody>
</table>
Note: We output one extra piece of information (in addition to our information fields) from our ASP page - that is "TotalRecords." TotalRecords is a numerical variable holding the number of records in the address book. This will help our Flash movie know when it has reached the end of our address book.

Our ASP page is called "GetDetail.asp" and will be saved in the same directory as our database and Flash files.

Flash Design

With our database and ASP page built to query it, we next to put together our Flash movie to produce the front end to our address book. Let's start off with a new movie and insert a blank movie clip into it.

The movie clip is going to be our address book, and it will consist of five text fields (for us to display our information in) and two buttons (left and right arrows used to navigate through the records.) The address book looks a bit like this:

The text fields have been created as dynamic text and have each been given a variable name. This will allow us to control their contents from within ActionScript.

Look at what happens when the movie clip is first loaded. We add an action to the clip to tell it to load up our ASP page with the first record as soon as it finishes loading. The ActionScript looks like this:

```actionscript
onClipEvent(load)
```
{ 
    CurrentRecord = 0;
    loadVariables ("getdetails.asp?Record=0", this);
}

It simply initializes our CurrentRecord variable (which we will use to keep track of our position in the address book) and then loads our GetDetails.asp page, which asks for the first record (i.e., Record 0).

One of the features of the loadVariables function is that it does its stuff asynchronously. This means that after Flash has executed a loadVariables command, it doesn't hang around waiting for the results to come back. Therefore, the data hasn't necessarily been loaded by the time the program reaches the line following the loadVariables function. So we need a mechanism to tell the movie to update our text fields whenever the data has finally loaded in. To achieve this we use the onClipEvent(data) action. This action is called whenever Flash has finished loading a set of variables. Our ActionScript looks like this:

onClipEvent(data)
{
    strName = Name;
    strTelephone = Telephone;
    strCity = City;
    strNotes = Notes;
    strPosition = "Record " add String(CurrentRecord+1) add " of " add String(TotalRecords);
}

This code simply transfers the variables retrieved from the ASP page into the text boxes that we added to our movie clip. It also updates a text field to show which record we are currently displaying.

Finally, we need to assign actions to the left and right arrows so we can navigate through the address book. Here's the code for the right (move to next record) arrow:

on (release)
{
    CurrentRecord++;
    if (CurrentRecord == TotalRecords)
        CurrentRecord = 0;
    loadVariables ("getdetails.asp?Record=" add String(CurrentRecord), this);
}

This code increases the CurrentRecord variable by 1 and checks to see whether we have gone past the last record in the address book. If we have, CurrentRecord is reset to 0, and the user is sent back to the first record in the address book. The code next loads the variables associated with the record from the ASP page. When the record has been loaded, Flash will call the onClipEvent(data) action again, and this will update the text boxes our users see.

The code for the left (move to previous record) arrow is virtually the same, except that we are decreasing the current record rather than increasing it.

And that's all we need to do. When we launch the SWF file from a browser, it will load the first record
into Flash variables within the onClipEvent(load) action. After the variables have been loaded, Flash will call the onClipEvent(data) action, where we update our text fields to display the information to our user.

Clicking on either navigation button will trigger actions that retrieve our ASP page and load in the new record, again calling on the onClipEvent(data) action. Here is how the final product looks:

![Image of Flash interface displaying contact information]

Conclusion

By combining the capabilities of Flash and ASP it's possible to create solutions that unite the graphical appeal of Flash with the data-retrieval capabilities of ASP. In this demonstration, we showed how it is possible to connect a Flash movie to an Access database. However, modifications to the ASP code would allow us to connect to an SQL server or any other data source for that matter.

About the Author

Matt Duckhouse lives and works in the United Kingdom. At DCS Automedia, the eBusiness solutions division of DCS Automotive, he develops Internet solutions for the automotive industry. Currently, he is working on a number of computer-based training systems, many of which are delivered over the Web using Flash to improve the presentation, while still maintaining content in a database. He can be reached at mattduckhouse@yahoo.co.uk.