

I was asked to write this tutorial after posting some code on how to get active slide show to work with asp.net. Actually it is easier than you would expect. In the following tutorial I'll explain on a step by step basis the approach. I haven't seen a cooler slide show, except for the slide show on an apple Macintosh, which looks exactly the same as Active Slide Show Pro (and also uses the same Ken Burns effects). But the benefit of Active Slide Show Pro is that you can use it on your website and let the whole world enjoy your slideshows. Feel free to ask any questions if you cannot get things to work.

An example can be found on <http://www.mijnvakantievrienden.nl> In this example I show 20 random images out of the database, combined with text at the top and per slide a link to another page.

Requirements for this tutorial.

I think anybody who is interested in this tutorial should have some knowledge of asp.net and also knows how to add data to a database and how to retrieve data from a database. So we start out with the basis of a website that already has images with their name in a table of the database.

In this sample I used:

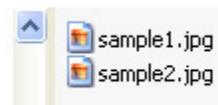
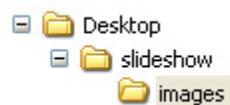
- Asp.net 2.0 in combination with SQL server
- Dreamweaver MX or newer
- Active Slide Show Pro (ASSP)

On the matter of asp.net 2.0 or 1.1 there are not that many differences. For SQL server, or MYSQL, or Access there are quite some differences and if needed just post a question and I'll do my best to answer a.s.a.p.

Let's get started.

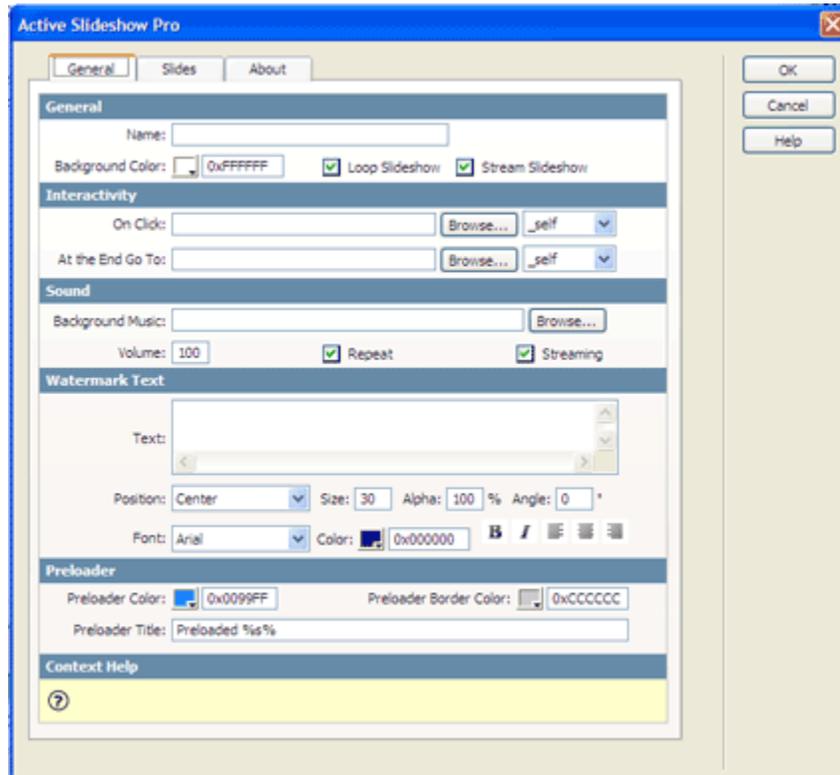
In this tutorial I have used the following folder structure;

- A folder on my desktop named slideshow
 - o in this folder another folder named images
 - with two images inside of the folder: sample1.jpg and sample2.jpg



First let's create a basic slide show with 2 slides, some text per slide and a link per slide. If you don't know how to do this take a look at the tutorial you can find on:
<http://www.dmxzone.com/ShowDetail.asp?NewsId=11360>

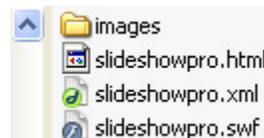
Click on:
 General workflow and slide options



Create the files below using the two images in the folder slideshow/images and save the html file in the slideshow folder as slideshowpro.html. The xml and swf are saved in the same folder by default.

After completing this task we should have three files:

- an html file
- an xml file
- a swf file



In my slideshowpro.html file I have:

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab
#version=7,0,0,0" width="450" height="300" id="test" align="middle">
  <param name="FlashVars" value="xmlFile=
slideshowpro.xml&preloaderBackColor=0x0099FF&preloaderBorderColor=0xCCCCCC" />
  <param name="movie" value="slideshowpro.swf" />
  <param name="quality" value="best" />
  <param name="scale" value="noscale" />
  <param name="wmode" value="window" />
  <param name="bgcolor" value="#FFFFFF" />
  <embed src="slideshowpro.swf" id="test" wmode="window" quality="best"
flashvars="xmlFile=
slideshowpro.xml&preloaderBackColor=0x0099FF&preloaderBorderColor=0xCCCCCC"
scale="noscale" bgcolor="#FFFFFF" type="application/x-shockwave-flash" width="450"
height="300" pluginspage="http://www.macromedia.com/go/getflashplayer" />
</object>
```

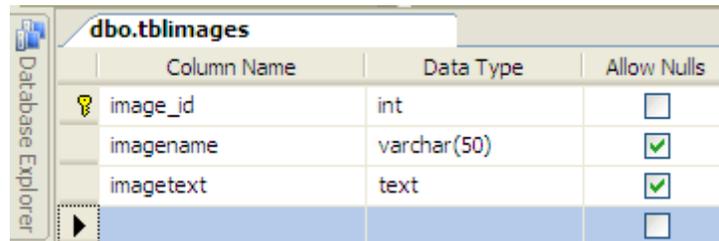
My slideshowpro.xml file:

```
- <slideshow loop="true" stream="true">
- <slide url="images/sample1.jpg" duration="3" fill="squeeze" background="0x000000">
  <transition effect="fade" position="0" easing="Linear" duration="2" />
  <kenburns panstart="9" panend="9" zoomstart="1.3" zoomend="1.3" />
  <click url="1.html" target="_self" />
  - <text position="0" font="Arial" color="0xFFFFFFFF" size="30" bold="false" italic="false"
align="center" alpha="100" rotation="0" delay="0" duration="3" inEffect="FadeIn"
inEasing="Linear" inDuration="1" outEffect="FadeOut" outEasing="Linear" outDuration="1">
    <![CDATA[ text with my first slide ]]>
  </text>
</slide>
- <slide url="images/sample2.jpg" duration="3" fill="squeeze" background="0x000000">
  <transition effect="fade" position="0" easing="Linear" duration="2" />
  <kenburns panstart="9" panend="9" zoomstart="1.3" zoomend="1.3" />
  <click url="2.html" target="_self" />
  - <text position="0" font="Arial" color="0xFFFFFFFF" size="30" bold="false" italic="false"
align="center" alpha="100" rotation="0" delay="0" duration="3" inEffect="FadeIn"
inEasing="Linear" inDuration="1" outEffect="FadeOut" outEasing="Linear" outDuration="1">
    <![CDATA[ text with my second slide ]]>
  </text>
</slide>
</slideshow>
```

My slideshowpro.swf is as it is....

Now we are going to start with the interesting part... How do we get our images not from a folder, but dynamically from our database. Actually there is no rocket science involved.

In this example I have a table in the database named tblimages with columns; imagename and imagetext.



Column Name	Data Type	Allow Nulls
image_id	int	<input type="checkbox"/>
imagename	varchar(50)	<input checked="" type="checkbox"/>
imagetext	text	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

All we want is to create a file similar to the xml file above, that is read by the file slideshow.swf. Maybe this needs some extra explanation, because I find the concept of a page that generates something that does something else... a little confusing. Anyway what happens is we have an html file where we call the slideshow swf file. The swf file needs a feed from another file which should be in the format of an xml file. In this tutorial we'll call an aspx file that generates the xml which can be read by the shockwave flash file.

Let's create a new apsx file and call it slideshow.aspx. In order to do this we could use Visual studio.net or Visual webdeveloper which is free but notepad could do the same trick with no problem at all or off course in Dreamweaver.

In my slideshowpro.html the only thing I adjust is the text slideshowpro.xml to slideshowpro.aspx. In slideshowpro.aspx (see the code below). I'm actually going to create the xml code based on the content in my database, dynamically.

```

10 <param name="FlashVars"
11 |value="xmlFile= slideshowpro.aspx&preloaderBackColor=0x0099FF&preloaderBorderColor=0xCCCCCC" />
12 <embed src="slideshowpro.swf" id="test" wmode="window" quality="best"
13 flashvars="xmlFile= slideshowpro.aspx&preloaderBackColor=0x0099FF&preloaderBorderColor=0xCCCCCC"
14 scale="noscale" bgcolor="*FFFFFF" type="application/x-shockwave-flash" width="450" height="300"
15 pluginspage="http://www.macromedia.com/go/getflashplayer" />
16

```

To get things to work the way we want them to. I'll give an example in codebehind but if somebody needs codebeside and cannot figure out how that should work please send an email and I'll see what I can do.

slideshow.aspx

Let's try to explain what happens. In my slideshowpro.html I changed slideshowpro.xml to slideshowpro.aspx. The code below is executed when slideshowpro.html is opened. The only difference is that I'm now producing the xml dynamically. The aspx page is executed and produces an xml page which is read by my slideshow.swf. In the code below; response.write does the trick. For example if I would use:

```
Response.write("dmxzone")
```

The result in my page that was produced would be: dmxzone.

Let's get started...

```
<%@ Page Language="VB" AutoEventWireup="false"
CodeFile="slideshow.aspx.vb" Inherits=" slideshow " %>

slideshow.aspx.vb

Imports System.Data.SqlClient
Imports System.Data
Imports System.IO
Partial Class slideshow
Inherits System.Web.UI.Page

Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

Dim txtimagename, txtimagetext As String

    Dim ConnStr As String =
System.Configuration.ConfigurationManager.AppSettings("MyProvider")
    Dim Sql As String

    Sql = "select imagename, imagetext from tblimages"

    Dim conn As New SqlConnection(ConnStr)
    Dim objDR As SqlDataReader
    Dim Cmd As New SqlCommand(Sql, conn)
    conn.Open()
    objDR = Cmd.ExecuteReader()
    Response.Write("<?xml version=""1.0"" encoding=""utf-8""?>" &
vbLf)
    Response.Write("<slideshow loop=""true"" stream=""true"">" &
vbLf)
    While objDR.Read()
        txtimagename = objDR("imagename")
        txtimagetext = objDR("imagetext")

Response.Write("<slide url=""images/" & txtimagename & "" duration=""5""
fill=""squeeze"" background=""0x000000"">" & vbLf)

Response.Write("<transition effect=""fade"" position=""0""
easing=""Linear"" duration=""2"" />" & vbLf)
```

```
Response.Write("<kenburns panstart=""9"" panend=""9"" zoomstart=""1.3""  
zoomend=""1.3"" />" & vbLf)  
  
Response.Write("<click url=""home.aspx"" target=""_self"" />" & vbLf)  
  
Response.Write("<text position=""6"" font=""Arial"" color=""0xFFFFFFFF""  
size=""10"" bold=""false"" italic=""false"" align=""left"" alpha=""100""  
rotation=""0"" delay=""0"" duration=""7"">" & vbLf)  
  
Response.Write("<![CDATA[" & txtimagetext & "]]>" & vbLf)  
  
Response.Write("</text>" & vbLf)  
  
Response.Write("</slide>")  
  
End While  
  
Page.DataBind()  
  
objDR.Close()  
  
conn.Close()  
  
Response.Write("</slideshow>")  
End Sub
```

So in the code below I reproduce the same xmlcode as ASSP generates except I generate it from my database.

What I basically do is the same what you would do if you want to generate a list from a table in the database.

Let's take a closer look at the next line of code.

```
Dim ConnStr As String =  
System.Configuration.ConfigurationManager.AppSettings("MyProvider")
```

You could have used:

```
Dim ConnStr As String = "SERVER=myserver;  
database=mydatabase;trusted_connection=yes"
```

All I do is refer to a setting in my webconfig file for the connection string to the database.

```
<appSettings>
  <add key="MyProvider"
value="server=myserver;database=mydatabase;trusted_connection=yes"/>
</appSettings>
```

In your projects use the above. When changing from a local project to a live project, you only have to change your connection string once instead of in every page

Before I start my while loop I first print the start of my xml page.

```
Response.Write("<?xml version=""1.0"" encoding=""utf-8""?>")
Response.Write("<slideshow loop=""true"" stream=""true"">")
```

These are the same lines that were generated by ASSP except I use response write to print them in my xml page.

In my while loop I produce the code for the lines for every slide in our database:

```
Response.Write("<slide url=""images/" & txtimagenam & "" duration=""5""
fill=""squeeze"" background=""0x000000"">" & vbLf)
Response.Write("<transition effect=""fade"" position=""0""
easing=""Linear"" duration=""2"" />")
Response.Write("<kenburns panstart=""9"" panend=""9"" zoomstart=""1.3""
zoomend=""1.3"" />" )
  Response.Write("<click url=""home.aspx"" target=""_self"" />" )
Response.Write("<text position=""6"" font=""Arial"" color=""0xFFFFFFFF""
size=""10"" bold=""false"" italic=""false"" align=""left"" alpha=""100""
rotation=""0"" delay=""0"" duration=""7"">" )
Response.Write("<![CDATA[" & txtimagetext & "]]>" )
Response.Write("</text>" )
Response.Write("</slide>")
```

Again this is the same code as the code that was generated in the xml file from ASSP except in this case we have two variables from the database.

After the loop I finish with:

```
Response.Write("</slideshow>")
```

In this sample I only used imagename and imagetext to retrieve from the database, but you could extend this to all your needs. Like; click url, transition effect, etc.

Please feel free to ask whatever question you have and I'll be happy to help.

Andy