XML Flash Slideshow v3

Tutorial - Creating dynamic data files

Contents

Getting Started

The first step in creating a dynamic data source file is to create the file. To do this, go to File > New… in Dreamweaver. In the left section, select the XML Flash Slideshow v3 item. Select the template appropriate for your server model and click Create below.
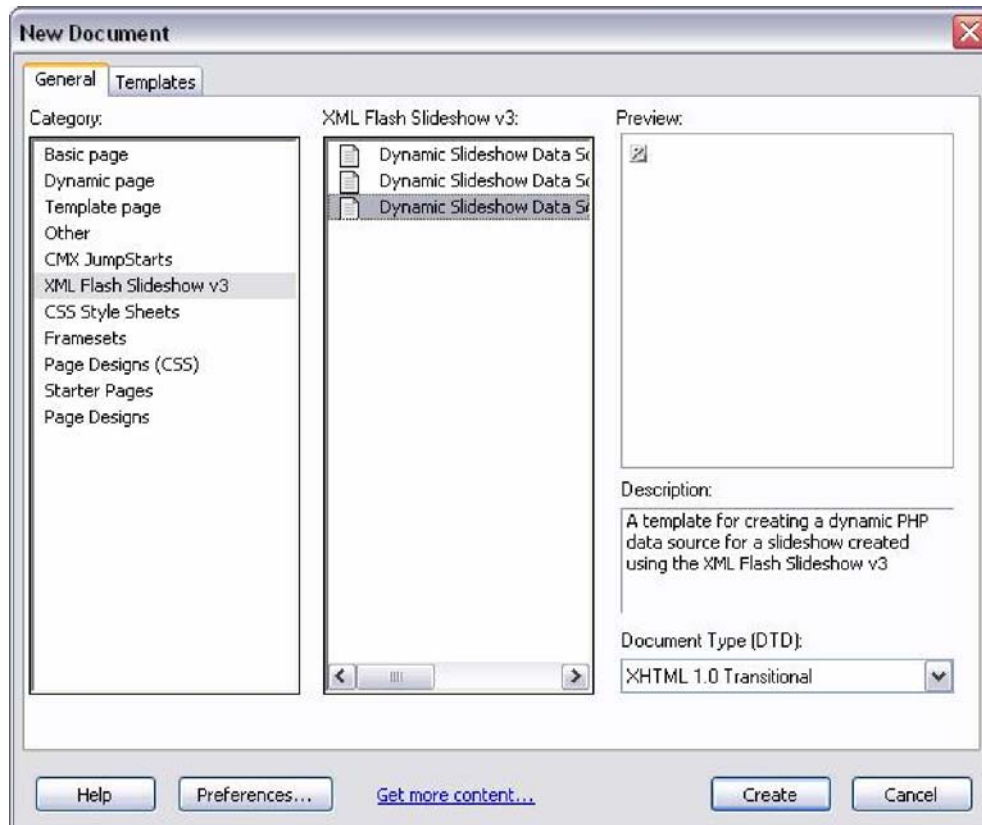
Save this file inside your Dreamweaver site. You will later use the URL of this file in the Dynamic Data section of the slideshow Insert Wizard.

Next, switch to code view in Dreamweaver. This tutorial is based on the PHP server model, but all code will be similar for other server models. You may see <!--DW6--> or some other code at the top of the file. Remove this code and any whitespace, up to the <?php code. After doing this, your file should look similar to the following:

```
1   <?php
2   /*
3       --- XML Flash Slideshow v3 ---
4
5       This is a template for a dynamic data source.  To use it, first create a recordset,
6       or create some other source of data.  Then, replace the values (REPLACE_WITH_URL, etc.)
7       with appropriate data items from the Bindings tab of the Application panel.  Next,
8       select the <img ... /> section of the page, and apply a 'Repeat Region' loop to this
9       section.
10
11  */
12  header("Content-type: text/xml");
13  echo("<?xml version=\"1.0\" encoding=\"utf-8\"?>");
14  ?>
15  <gallery>
16
17
18      <!-- Start repeat region here -->
19
20
21      <img src="REPLACE_WITH_URL" title="REPLACE_WITH_TITLE" caption="REPLACE_WITH_CAPTION" link="REPLACE_WITH_LINK" />
22
23
24      <!-- End repeat region here -->
25  </gallery>
26
```

Your file is now ready to have dynamic data added.

The Dynamic Data XML Format

When your file is finished, you should be able to preview the file in a browser, and see an XML document. The format of this XML is very important, to ensure that the XML can be properly read by your slideshow. The end result must be in this format:

```
<gallery>
        <img src="images/image1.jpg" title="Image 1 title" caption="" link="" />
        <img src="images/image2.jpg" title="Image 2 title" caption="" link="" />
        <img src="images/image3.jpg" title="Image 3 title" caption="" link="" />
</gallery>
```

There is a <gallery> node which holds multiple <img /> nodes. Each <img /> node represents a single image. Each attribute specifies the following:
- src="" portion represents the URL of the JPEG image (required)
- title="" portion represents the title for the image (optional)
- caption="" portion represents the caption for the image (optional)
- link="" portion represents the hyperlink for the image (optional)

About Integrating the Dynamic Data

When you create your file, you must start with a dynamic list of images. There are many different sources for a list like this, including a Recordset (most used), an array of files in a directory, or a static list of images that you have randomized. More details about these different sources will be mentioned later on.

To translate this list to the proper XML format, you must create a loop in your dynamic data document. This loop will go over the image list, and output an `<img src="" />` node for each of these items.

The places where the beginning and end of this loop should go are denoted by `<!-- Start repeat region here -->` and `<!-- End repeat region here -->` in the dynamic data file template.

Specific Example - Using a Database Recordset

In this specific example, it will be shown how to use database data in your dynamic data file.

First, you must use the Application panel to insert a recordset into the file. If you are not familiar with creating a Database connection and a Recordset, you will need to first learn basic Dreamweaver dynamic data concepts. To learn about this type of Dreamweaver development, see: http://livedocs.macromedia.com/dreamweaver/mx2004/using/part8_dy.htm .

Now, switch to the Bindings section of the Application panel. Expand your recordset, so you can see each of the recordset fields:
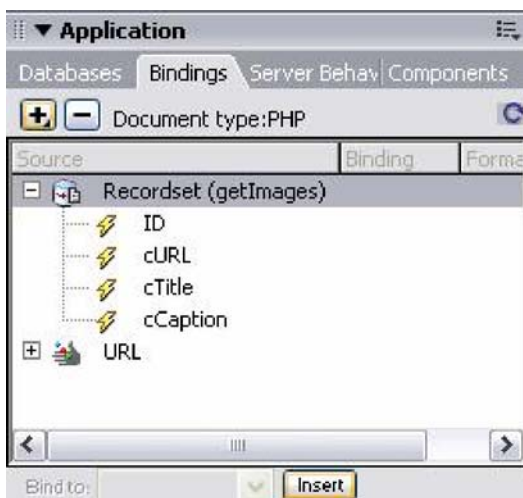
In this example, the recordset has been named getImages. The database table has three relevant columns – cURL, cTitle, and cCaption. In the document's code, locate the <img /> line, and select the src attribute:

```
<img src="REPLACE_WITH_URL" title="REPLACE_WITH_TITLE" caption="REPLACE_WITH_CAPTION" link="REPLACE_WITH_LINK" />
```

Back in the Bindings panel, select the recordset field which represents the image URL (cURL in this example), and drag it onto the REPLACE_WITH_URL selection. It should replace the current contents, leaving you with something like this:

```
<img src="<?php echo $row_getImages['cURL']; ?>" title="REPL
```

If you want to use a title, caption, or link from the database, replace the other attributes with data from the bindings panel. If you don't want to use these alternate attributes, simply remove them.

The <img /> section of your document should now look similar to this:

```
<!-- Start repeat region here -->

<img src="<?php echo $row_getImages['cURL']; ?>" />

<!-- End repeat region here -->
```

You are now ready to setup the looping for the <img /> node. Select the entire line, and switch back to the Application Panel. Select the Repeat Region option:
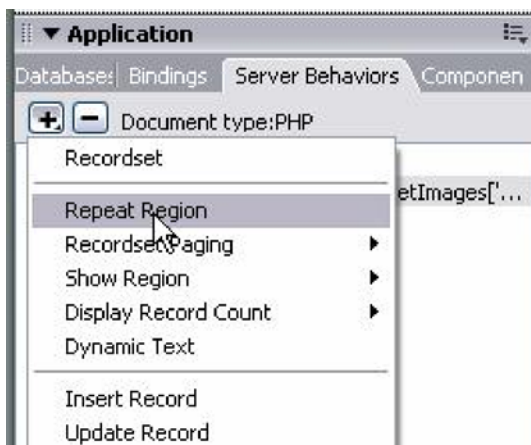
Select the appropriate recordset, and select the 'All Records' option. Click OK.

Some new code will be added to your document. It should now look similar to this:

This loop will output an <img /> node for each record in the database recordset. You are now ready to preview the file, and make sure that everything is properly configured. Upload the page, and open it in the browser. You should see something like this:
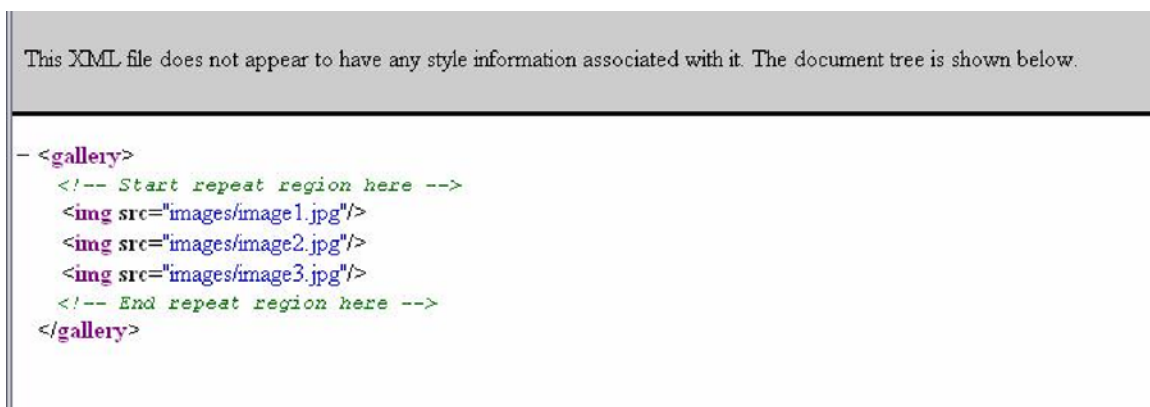
If you instead see a PHP error about 'Output already started at line...', then make sure that you have no whitespace before the very first <?php code block. Once you can see output as is shown above, then congratulations - you've successfully created the dynamic data source.

Specific Example - Listing all images in a directory

To use all images in a directory, you first need to get the list of all of the images. To begin, create a new PHP code block above the <gallery> node:

Inside this new code block, add the following code:

```php
// Specify a directory
$dir = "../images/";

// Create a holder array
$images = array();

// Get the list of images
if ($dh = opendir($dir))
{
        while (($file = readdir($dh)) !== false)
        {
                // Make sure that this is a JPEG image
                if (preg_match("|\.jpe?g$|i", $file) == false)
                {
                        continue;
                }

                // Add the image to the image array
```

```
                    $images[] = $dir . $file;


            }
        closedir($dh);
}
```

This code will get all of the JPEG images in the directory specified in red. You must replace the red text with the path to your directory. An $images array is created, and filled with the image URL's.

Now, you're ready to add looping code. Above the <img /> node, add the following:

```
<?php
foreach ($images as $imageURL)
{
?>
```

Directly below the <img /> node, add this:

```
<?php
}
?>
```

Your code should now look like this:

```
<gallery>
|

    <!-- Start repeat region here -->

    <?php
    foreach ($images as $imageURL)
    {
    ?>

    <img src="REPLACE_WITH_URL" />

    <?php
    }
    ?>

    <!-- End repeat region here -->

</gallery>
```

Now that the loop is set up, replace the REPLACE_WITH_URL text with the following code:

```
<?php echo($imageURL); ?>
```

Your <img /> line should now look like this:

```
<img src="<?php echo($imageURL); ?>" />
```

You can now preview the file in your browser. You should see something similar to this:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
- <gallery>
    <!-- Start repeat region here -->
    <img src="../images/photo1.jpg"/>
    <img src="../images/photo2.jpg"/>
    <img src="../images/photo3.jpg"/>
    <img src="../images/photo4.jpg"/>
    <img src="../images/photo5.jpg"/>
    <!-- End repeat region here -->
</gallery>
```

If you see something like this, then you have successfully created the script. All of the JPEG images in the directory have been read and outputted by the dynamic data source.

Specific Example - Randomizing a Static List of Images

In this example, you will start with a preset list of image URL's you want to use in the slideshow gallery. Then, you will randomize the order of this list, so that the user will always see the images in a random order.

To begin, create a new PHP code block above the <gallery> node:

```
.6        section.
.7
.8    */
.9    header("Content-type: text/xml");
:0    echo("<?xml version=\"1.0\" encoding=\"utf-8\"?>");
:1    ?>
:2
:3
:4 ⊟ <?php
:5  |
:6 ⊟ ?>
:7
:8    <gallery>
:9
:0
:1        <!-- Start repeat region here -->
```

Inside this new code block, add the following code:

```php
$images = array("images/image1.jpg", "images/image2.jpg", "images/image3.jpg");
shuffle($images);
```

This code creates an $images array holding image URL's. You must specify all of your image URL's in this array. Then, it randomizes the order of the array. Now, you're ready to add looping code. Above the <img /> node, add the following:

```php
<?php
foreach ($images as $imageURL)
{
?>
```

Directly below the <img /> node, add this:

```php
<?php
}
?>
```

Your code should now look like this:

```
<gallery>

    <!-- Start repeat region here -->

    <?php
    foreach ($images as $imageURL)
    {
    ?>

    <img src="REPLACE_WITH_URL" />

    <?php
    }
    ?>

    <!-- End repeat region here -->

</gallery>
```

Now that the loop is set up, replace the REPLACE_WITH_URL text with the following code:

```
<?php echo($imageURL); ?>
```

Your <img /> line should now look like this:

```
<img src="<?php echo($imageURL); ?>" />
```

You can now preview the file in your browser. You should see something similar to this:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
- <gallery>
    <!-- Start repeat region here -->
    <img src="../images/photo1.jpg"/>
    <img src="../images/photo2.jpg"/>
    <img src="../images/photo3.jpg"/>
    <img src="../images/photo4.jpg"/>
    <img src="../images/photo5.jpg"/>
    <!-- End repeat region here -->
  </gallery>
```

If you see something like this, then you have successfully created the script. Each time you reload the page, a new image order should appear.