

How to Create a Dynamic RSS Reader

Audience: web developers (intermediate and advanced only)

Task

RSS feeds are very useful within sites, as they allow integrating foreign content in an automatic manner. Just select the source for the RSS feed, and when the original content changes, it will reflect in the content you find on your site. For more information concerning RSS, consult this page <http://www.w3.org/2001/10/glance/doc/howto>.

InterAKT's MX RSS Reader-Writer can help you integrate a RSS feed into one of your pages. This will display the same feed to all of your site visitors. However, you can use some tricks to retrieve a specific feed in a dynamic manner and allow your visitors to select the feed they want.

Scenario

The main purpose of this tutorial is to create a site that allows users select the RSS Feed that they want to see in the page from a drop-down menu, or even enter their own in a text box. Both in design and functionality, the site will resemble InterAKT's live demo pages for the MX RSS Reader-Writer. The existing live demo can be viewed here:

<http://www.interaktonline.com/demos/MXRSS/>

We assume you have completed these steps:

- You have properly set up your application development environment.
- You have already installed the **MX RSS Reader-Writer** Dreamweaver extension.
- You have properly defined the site in Macromedia Dreamweaver MX using one of the supported server models: **PHP_MySQL**, **PHP_ADODB**, **ColdFusion**, or **ASP VBScript**.
- You have created the site pages and have set up a successful database connection.
- You already have knowledge of configuring the **MX RSS Reader-Writer** dialog box.

Solution

You can easily display dynamically selected feeds just by customizing the code that calls the RSS reader function.

The solution presented here was designed for **PHP**. For **ASP** and **ColdFusion**, the approach is similar.

In order to create a nicely looking dynamic page, you will have to define CSS styles for formatting your RSS feeds. To do so, follow the next steps:

1. Open the page which will display the feed into Dreamweaver. The simplest approach to take in order to define the styles to use in the page, and at the same time to keep a low number of files is to add the style definition on top of the page. You can either use Dreamweaver's built-in CSS editor, or just type the code by hand. The styles created here should cover the body section of the page, redefining the font size, family and color, as well as links, table cells, headers and input elements.

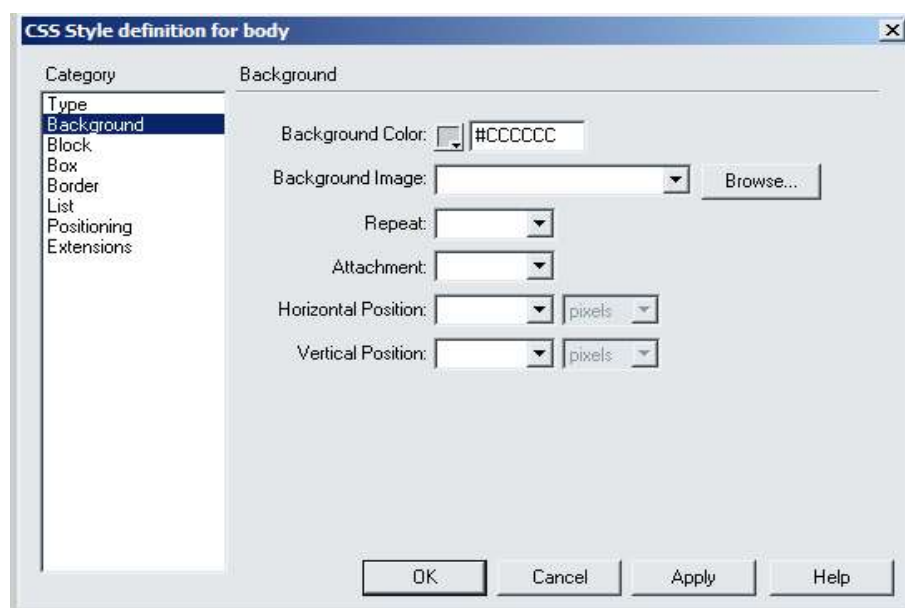


Figure 1 Define new styles for page elements

2. Since CSS coding isn't in the scope of this document, you can find below the actual code to include in your page. This will make your page look similar to the live demo page:

- For the body and links style, the definitions look like:

```
body {  
    padding-right: 10%;  
    padding-left: 10%;  
    background-color: #FFFFFF;  
}
```

```
a {  
  text-decoration: none;  
}
```

- The form which will allow users select the desired feed should have a distinctive style also:

```
input {  
  background-color: #e0f0ff; border: 1px solid  
#333333;  
  color: #333333;  
}  
select {  
  background-color: #e0f0ff;  
  border: 1px solid #333333;  
  color: #333333;  
}
```

- Table elements (cells, rows, headers) as well as channel title and description have their own styles too:

```
.tableStyle { background-color: #FFFFFF;}  
.tableHead {  
  color: #FFFFFF;  
  background-color: #3d4171 !important;  
  height: 120px;  
  text-align: left;  
  padding-left: 5px;  
  padding-top: 10px;  
}  
.tableRow {  
  background-color: #EAEAEA;  
  border: none !important;  
  padding-left: 10px;  
  padding-right: 10px;  
}  
.channelTitle {  
  border-bottom-width: 3px;  
  border-bottom-style: solid;  
  border-bottom-color: #FF9933;  
  color: #3d4171 !important;  
  font-size: 18px;  
  line-height: 40px;  
}  
.channelDescription {  
  font-size: 11px !important;  
}  
.tableHead form {  
  text-align: right;  
  font-size: 12px;  
  vertical-align: middle;}
```

3. The next logical thing to do is to add a RSS Feed to the page. You can do this easily, by using the **MX RSS Reader** command. You can find it as an icon on the **InterAKT** tab of the **Insert bar**. Just click the icon and fill in the simple dialog box, with your connection, table and a **RSS** feed. The one used in the making of this tutorial is:

<http://www.w3.org/2000/08/w3c-synd/home.rss>

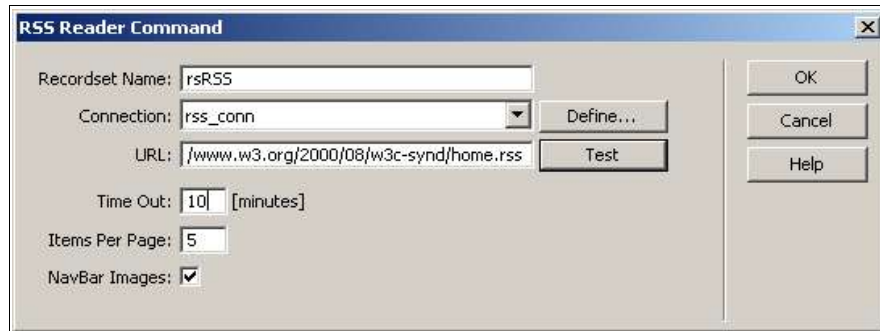


Figure 2 The RSS Reader command dialog box

4. Since style is one of our goals, check the *NavBar images* option. Once you click the **OK** button, the page's layout will be automatically created, displaying the channel's title, a description, the navigation bar and the actual content. The layout is organized as a table, and as such, inserting data or moving it around is reduced to creating or deleting new rows:

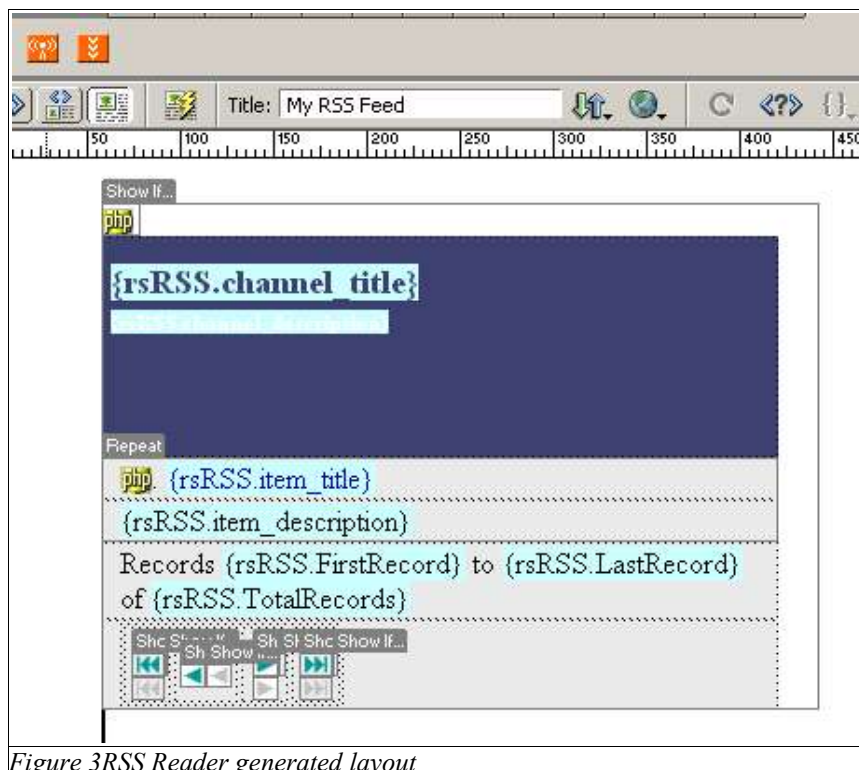


Figure 3 RSS Reader generated layout

5. As you can see, one of this tutorial's goals has already been reached: the page containing the RSS Feed looks similar to the one used for the Live Demo. Next, you must add elements that will allow the users select from one of the offered feeds, or enter their own. And since the only way for users to input data into a web page is through forms, this is what will be created next.
6. Make some modifications to the table's header so that a logo and the form elements can be displayed. Right click in the blue area (the table header) and choose **Table-> Insert row**. This way, a new row has been inserted above the one containing the channel title and description. The problem is that we now have two table headers. To correct this, click in the second row of the table, and, from the **Property inspector** un-check the *header* option. You'll also have to change its class from *tableheader* to *tablerow*, to avoid having two blue areas. Right click on the `<td>` tag from the **Property Inspector** and select **Set class-> tablerow**. The row should have turned gray by now.
7. It's time to create the new header. Click in the first table row and insert a logo. For this tutorial, the **MX RSS Reader-Writer** logo has been used. To insert an image, open the **Common tab** of the **Insert Bar** and click on the image icon. A dialog box will request you to select the image file, and upon confirmation, it will insert it into the page.
8. The second task regarding the header is to add the input elements to allow users select the desired RSS Feeds, or enter one of their own. In this step you should not concern yourself with the actual code or values to enter. Just add the elements to use: a form, a Jump Menu, a text box and a Submit button. All of these elements can be found on the **Forms tab** of the **Insert bar**. Do not add any values for the Jump Menu at this time. Just click the **OK** button to add it to the page. Set the text field's name to *rss_text*, and the form's method to **POST**. You do not have to enter any value for the action property, as the page will submit to itself.
9. To be able to continue and convert the page to a dynamic RSS reader, you must first understand how is the URL passed to the page. To see this, switch to **Code view** in Dreamweaver. The lines of code to look for are similar to the following:

```

$MXRSS_rsRSS = new MX_RssReader();
$MXRSS_rsRSS->setConnection($rss_1);
$MXRSS_rsRSS->setCacheTime(10);
$MXRSS_rsRSS->getFeed("http://www.w3.org/2000/08/w3c-synd/home.rss");

```
10. The code may vary depending on the options selected in the **RSS Reader** dialog box. The line of most interest right now is the last one, as it defines the feed to use. As you can see above, it is set in a static manner, but is nothing more than a text variable, after all. Taking this into consideration, the logical thing to do in order to turn it dynamic is to replace the static

text variable by a variable we can change easily: a **POST** variable. The choice of **POST** variables is easy: they are actually the names of form elements, which are also the ways for users to input data and they change as the page is submitted. This way, the feed URL can be entered by the users in the text box.

11. Replace the static feed URL with `$_POST['rss_text']`. This will set for URL the value of the `rss_text` form field. There is however a trap: if the page is called without entering an URL into the text field, an error will be triggered. To avoid this, a check should be made on the variable. If it isn't sent, then it should be initialized with a default URL. Add the following lines of code at the very top of the page:

```
<?php if (!isset($_POST['rss_text'])) {  
    $_POST['rss_text'] =  
    "http://newsvac.newsforge.com/newsvac.rss";  
}
```

12. The code above tells the page that if the `rss_text` hasn't been set, then it should automatically display the newsforge feed.
13. With this, the feed has become dynamic. All that remains is to link the menu inserted earlier to the `rss_text` text field. This way you can create a list of feeds that your site visitors will be able to choose from. The modification required is of **JavaScript** nature. You will have to change the javascript code behind the menu so that instead of opening a new page, it will set the `rss_text` value. Select the menu and right click on the `<select>` tag in the Property inspector. Select Edit tag to change its behavior. Replace the existing `onChange` event with the following code:

```
onChange="document.forms['form1'].rss_text.value =  
document.forms['form1'].menu1.value; document.forms  
['form1'].submit();" 
```

14. The new `onChange` code will set the `rss_text` value to the one it has selected, and also submit the page. As you can notice, the code is made for a form called `form1`, and the menu's name is `menu1`. You can now add elements to the menu. Just enter a suggestive name in the text box and the URL in the value box.

15. The last thing to do is to save and preview your page. It should look just like MX RSS Reader-Writer's live demo page:



Other Resources

For more details, check out the other useful documents that come with MX RSS Reader-Writer:

- *Getting Started with MX RSS Reader-Writer*
- *The MX RSS Reader-Writer User Manual*