



Creating a Simple Blog Part 3

By: Tom Muck

One of the frequent questions in the forums is "How do I create a blog?". There are many commercial blog systems out there, and many full-featured blog providers, like Blogger, but you can also create a simple blog using the standard tools of Dreamweaver. This tutorial will be equally applicable to ColdFusion, ASP, PHP, ASP.NET, or JSP because we will be using standard Dreamweaver server behaviors to create the blog. Part 1 of the tutorial can be found at <http://www.communitymx.com/abstract.cfm?cid=7CC52> (<http://www.communitymx.com/abstract.cfm?cid=7CC52>). Part 2 can be found at <http://www.communitymx.com/abstract.cfm?cid=C6355> (<http://www.communitymx.com/abstract.cfm?cid=C6355>). Part 3 will focus on adding more features to the blog, such as an RSS feed, individual blog item pages, and how to automatically insert dates.

This series of articles will assume that you know how to work with Dreamweaver to create a data-driven site. For example, you should know how to create databases and connections within your chosen server model.

When we left off, we had a basic blog that allowed you to post items and display them, as well as edit them and post comments. In those pages -- the comments page in particular -- you had to manually type a date to insert it. An easier way is to let the database do the work. Because the first two parts of the series showed DW point-and-click functionality, we did not implement automatic dates. Now we'll describe how to do this.

Each database system has a specific function for retrieving a current date. Table 1 shows several popular databases and the function used:

Database	Function
MySQL	Now()
SQL Server	getdate()
MS Access	Now()
PostgreSQL	Now

Table 1: Databases and default current date function

You can implement default dates with your particular database system using a visual tool, such as the Enterprise Manager in MS SQL Server shown in Figure 1.

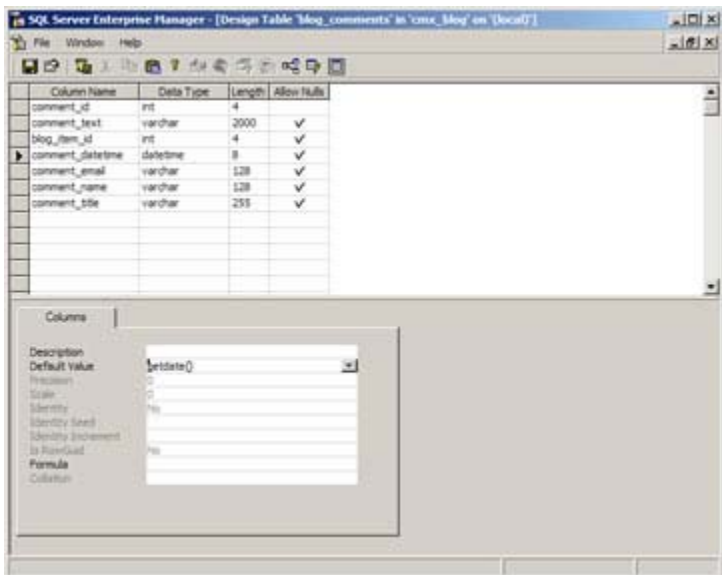


Figure 1: Adding a default in SQL Server Enterprise Manager

You can also code the default with a SQL statement to change your current table structure to add a default, like this:

```
ALTER TABLE blog_comments ADD
CONSTRAINT DF_blog_comments_comment_datetime
DEFAULT (getdate()) FOR comment_datetime
```

With that in place, you can modify your Dreamweaver insert statements so that the date is no longer inserted by the user. Simply open up the **comment.php** page (or .cfm, .asp, etc) and double-click the Insert server behavior that already exists in the Server Behaviors panel. Remove the entry for comment_datetime. Also remove the form field on the page that supplies the comment_datetime value. Now, the database will take care of this value.

Individual Blog Items

Your blog should have the ability for individual blog items to be linked to or otherwise referenced. One way to do this is by adding some conditional logic in the SQL statement using the scripting language of the page directly on the main index page, like this (pseudo-code):

```
IF URL.id NOT EQUAL TO "" THEN
    myFilter = " WHERE blog_item_id = " + URL.id
END IF

SQL = "SELECT * FROM blog_items " + myFilter
```

However this method is not very friendly to the Dreamweaver environment. For that reason, we'll simply create a separate page to display individual items. Create an **archive.php** page (or .cfm, .asp, etc) within the main blog site as a direct copy of the **index.php** page. By doing it like this, all of the formatting of the

blog items will be identical, and the recordset will already be set up. Now, add a filter to the recordset by opening it up in the Bindings panel and using the simple Recordset dialog box to add a filter that is a URL variable named **id**, as shown in Figure 2. You can also remove the ORDER BY clause from this dialog box as well by setting Sort to **None**.

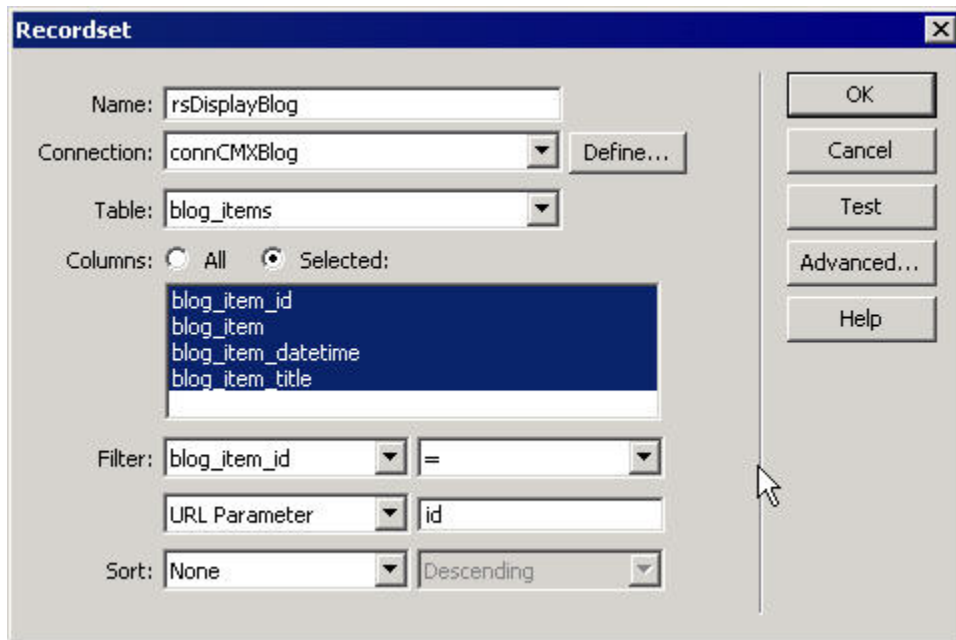


Figure 2: Simple recordset dialog box used to easily add filters

Now that we have an archive page, how is it referenced? Simple -- open up the **index.php** page and add a link to {rsBlogItems.blog_item_title} within the Repeat Region. Use the Property Inspector and follow these steps:

1. Select the title within the Repeat Region
2. Next to the Link field in the Property Inspector, use the point-to-file icon to drag the link to the archive.php page in the Files panel, as shown in Figure 3.

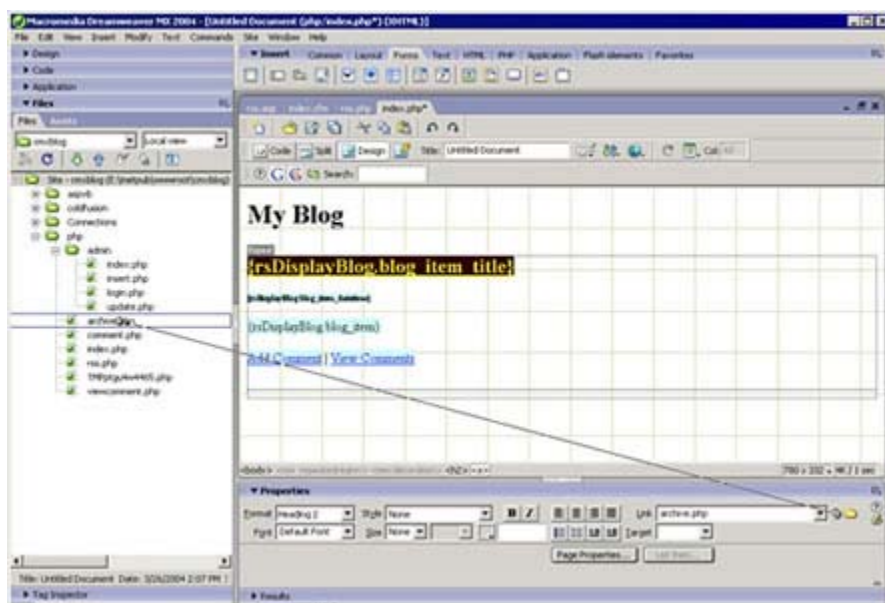


Figure 3: Linking to the archive page

3. Next, click the folder icon next to the Link field, then click the Parameters button to add a parameter to the link: it should be a URL parameter named ID using the dynamic data lightening bolt to choose the blog_item_id field as the value, as shown in Figure 4.

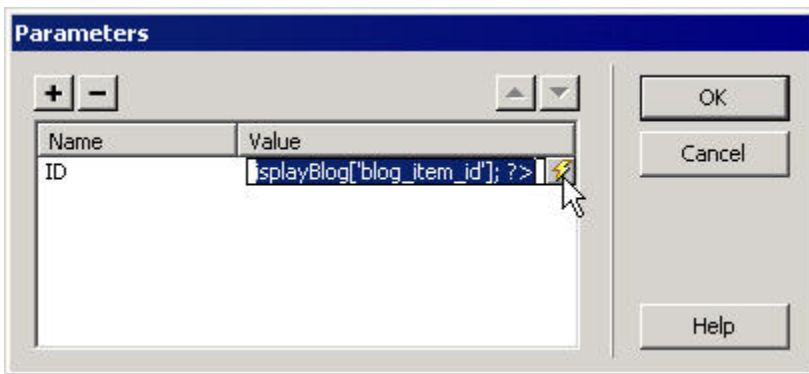


Figure 4: Adding a URL variable to a link

4. Save the page -- you now have a link for each blog item for search engines and other direct links. As a side benefit of copying the index page, all comment functionality will work from the archive page as well.

RSS Feeds

One item that you may want to add to your blog is an RSS (Really Simple Syndication) feed. A feed accomplishes one thing -- allows your content to be syndicated on other sites and applications, such as Macromedia's XML News Aggregator. An aggregator is a great place to find the latest content from many blogs without having to visit each one individually.

An RSS feed is a simple XML document that conforms to a predefined structure, usually RSS 0.92 or later (0.91 had a limit of 15 items). An RSS document looks like the following:

```

<?xml version="1.0" ?>
<!-- RSS generation by 'CommunityMX.com'
    on Sat, 10 Apr 2004 17:44:06 GMT -->
<!-- Last 30 entries at Community MX -->
<rss version="0.92">
  <channel>
    <title>CommunityMX.com</title>
    <link>http://www.communitymx.com</link>
    <description>Community MX -- Extending
Knowledge</description>
    <webMaster>admin@communitymx.com</webMaster>
    <language>en-us</language>
    <item>
      <title>A Simple Banner Application: Part 2</title>
      <description>In Part 1, we covered the creation of the
database, the application.cfm file, and a basic page to
display our banners. This, the final installment, will take
you through monitoring the number of times an ad is shown and
counting each click-through a banner
receives....</description>
      <link>http://www.communitymx.com/abstract.cfm?
    on cid=99ACC</link>
      <guid
isPermaLink="true">http://www.communitymx.com/abstract.cfm?
    on cid=99ACC</guid>
      <pubDate>Fri, 09 Apr 2004 05:00:00 GMT</pubDate>
    </item>
    <item>
      <title>Instant Replay Part 1--Fireworks</title>
      <description>No striped shirts or trips to the replay
booth are required when you use the Replay button in
Fireworks. Instead, you can make short work of repetitive
tasks, and even use the power of this nifty little feature
for some fascinating creative effects--all with the click of
a button. This 3-part video tutorial demonstrates the
creative uses of the History panel and the Replay button and
shows how you can use them to achieve some creative
results....</description>
      <link>http://www.communitymx.com/abstract.cfm?
    on cid=807A1</link>
      <guid
isPermaLink="true">http://www.communitymx.com/abstract.cfm?
    on cid=807A1</guid>
      <pubDate>Fri, 09 Apr 2004 05:00:00 GMT</pubDate>
    </item>
  </channel>
</rss>

```

The RSS feed has a channel tag, which contains `<title>`, `<link>`, and `<description>` tags within that describe the content. Optionally, you can have other tags as well, such as `<language>` and `<webmaster>` in the listing above, and `<managingEditor>`, `<image>`, `<lastBuildDate>`, and other tags not used in the example.

The `<item>` tags contain the actual content. You should limit the content in the RSS feed by only displaying the first 100 words or so from your blog item. Inside the `<item>` tags are required `<title>` and `<description>` tags, and optional `<link>`, and `<pubDate>` tags. These will have to be looped from the database, using some hand-coded scripting in your server language. The way we will do this is to build a separate page in the site that will generate the feed from the database and dynamically generate the tags.

We'll start with the skeleton of the RSS feed and fill it in with dynamic content:

```
<?xml version="1.0" ?>
<rss version="0.92">
  <channel>
    <title>{Blog title here}</title>
    <link>{Main blog link here}</link>
    <description>{blog description here}</description>
    <webMaster>{your admin email address here}</webMaster>
    <language>en-us</language>
    {start of loop over blog items}
    <item>
      <title>{title here}</title>
      <description>{content here}</description>
      <link>{link here}</link>
      <guid isPermaLink="true">{link here}</guid>
      <pubDate>{date here}</pubDate>
    </item>
    {end of loop}
  </channel>
</rss>
```

What we will want to do is to create a page that generates the content using this skeleton, but feeds it as a **text/xml** content type. The area within the item loop will come from the database. We'll try to do as much as possible with built-in Dreamweaver functionality:

1. Start with a blank page -- by "blank" I mean completely blank, none, nothing, nada. Add a recordset to the page using this query:

```
SELECT blog_item_id,
blog_item_datetime,
blog_item_title,
Left(blog_item,250) as blog_item
FROM blog_items
ORDER BY blog_item_datetime DESC
```

You can see that we are only getting the first 250 characters from the **blog_item** field. You can modify this as required, or use your own custom function here (such as one that will split words at the word boundary) or simply use the **blog_item** field with no modifier to display the entire item.

2. Next, add the skeleton to the page below the recordset. Fill in the tags within the `<channel>` tag that comes before the `<item>` tags with information that pertains to your blog.

3. Open the Bindings panel and expand the recordset. Drag the fields into the appropriate places within the `<item>` tag or click the Insert button on the Bindings panel, such as the **blog_item_datetime** field shown in **Figure 5**.

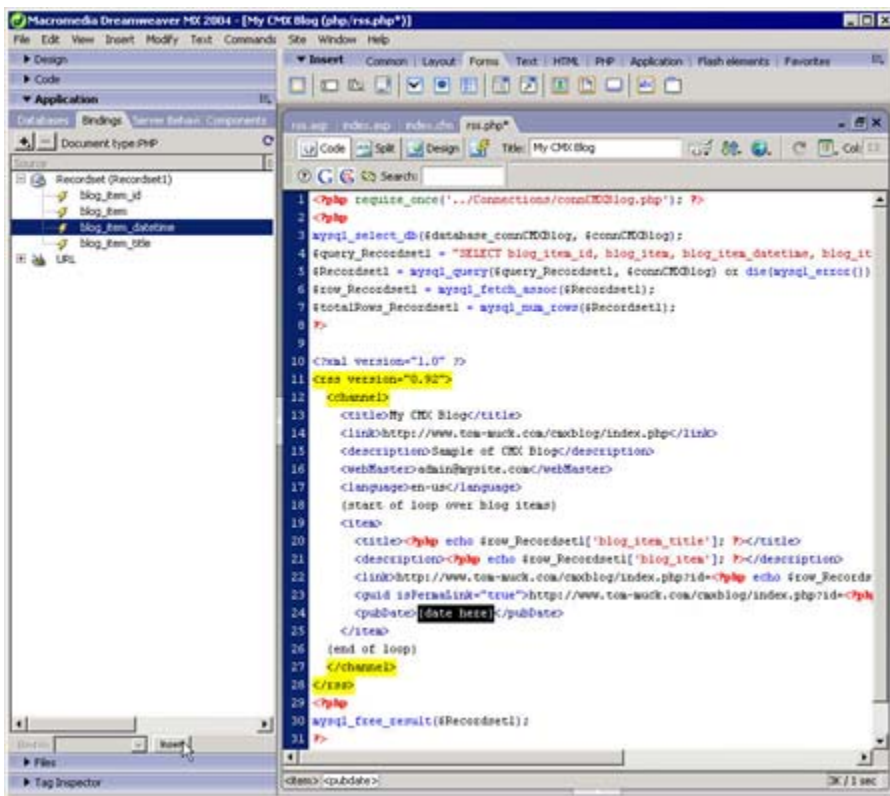


Figure 5: Inserting a field into the XML skeleton

4. The link and guid tags should contain fully qualified links to your blog item. For example, if your blog is at <http://www.jumpinjahosephat.com/>, your link urls should look like this, minus the ID:

```
http://www.jumpinjahosephat.com/archive.php?id=
```

5. Set up the link URL id by simply dragging the field from the Bindings panel to the cursor location directly after the **id=**

6. With all of the above items in place, your page will now generate a RSS feed.

There is one catch, however -- a browser will not pick it up as XML despite the XML declaration at the top of the code. You have to set the content type of the page. You can do this with a little bit of scripting directly before the output of the XML. Below is the code for the available server models (JSP and ASP.NET can have content type directly set in the `@page` directive):

PHP

```
<? header("Content-type: text/xml"); ?>
```

ASP

```
<% Response.ContentType = "text/xml" %>
```

ColdFusion

```
<cfheader name="content-type" value="text/xml">
```

JSP

```
<%@ page contentType="text/xml;charset=ISO-8859-1" %>
```

ASP.NET

```
<@ Page ContentType="text/xml" %>
```

7. Finally, the page will not validate as RSS without modifying the **blog_item_datetime** field to the proper format expected by RSS (using [RFC 822 specifications](http://www.faqs.org/rfcs/rfc822.html) (<http://www.faqs.org/rfcs/rfc822.html>)):

```
<pubDate>Tues, 13 April 2004 3:21:36 GMT</pubDate>
```

The format uses the GMT time, so a calculation has to be done on the date before displaying it in ColdFusion. You should find out your GMT offset and use that to perform the calculation. For example, Eastern Standard Time (US) is GMT -5, so I have to add 5 hours to the final date in the ColdFusion code. Other server languages have GMT conversions. Each server language has it's own date/time formatting functions, so the code is listed below. Note: the VBScript version uses a JavaScript function because there are no GMT date formatting functions in VBScript:

PHP

```
<?php echo date('r',$row_Recordset1['blog_item_datetime']);
?>
```

ASP

Function defined earlier:

```
<script language="JScript" runat="Server">
function convertToRSSEfriendlyDate(theDate) {
    var dt = new Date(theDate);
    var datestr = dt.toGMTString();
    datestr = datestr.replace(/UTC/, "GMT");
    return datestr;
}
</script>
```

within the <pubdate> tag:

```
<%=convertToRSSEfriendlyDate
    ↪(rsGetBlogItems("blog_item_datetime"))%>
```

ColdFusion

```
#dateFormat(dateadd('h',5,rsGetBlogItems.blog_item_datetime),
"ddd, dd mmm yyyy")#
#timeFormat(dateadd('h',5,rsGetBlogItems.blog_item_datetime),
"HH:mm:ss")# GMT
```

JSP

```
<% java.util.Date theDateObj =  
rsGetBlogItems.getDate("blog_item_datetime");  
java.text.SimpleDateFormat formatter = new  
java.text.SimpleDateFormat("EEE, d MMM yyyy HH:mm:ss Z");  
//java.util.Date theDateObj = new java.util.Date();  
String dateString = formatter.format(theDateObj);  
out.print(dateString);  
%>
```

ASP.NET

```
<%#  
DateTime.Parse(rsGetBlogItems.FieldValue("blog_item_datetime",  
Container)).ToString("r") %>
```

After building your RSS feed, you'll want to validate it. There are numerous validators available on the web, including this one for 0.92 feeds: <http://rss.scripting.com/> (<http://rss.scripting.com/>). If your feed does not validate, you may have to make adjustments in the content or format. For example, there should be no whitespace before the `<?xml version="1.0" ?>` declaration. In the ColdFusion version, a `<cfsilent>` tag comes in handy for suppressing whitespace.

Lastly, place a link somewhere in your blog to the new RSS feed, using the following format:

```
<a type="application/rss+xml" href="feed.php">My RSS Feed</a>
```

The `<a>` tag includes a type of `application/rss+xml` so that news aggregators can find it and pull the feed content. You can also include a link in the head section of the blog:

```
<link rel="alternate" type="application/rss+xml"  
href="feed.rss" title="My RSS Feed ">
```

Sample RSS feeds are included in the accompanying zip file.

Conclusion

Blogs are simple content management systems that can be put together easily using some built-in Dreamweaver functionality. In this part we've added default values for the date/time fields, an archive page, and added an RSS feed to the site so that news aggregators can syndicate the content.

Approximate download size: 332k

Keywords

blog,dreamweaver,server behaviors,coldfusion,jsp,java,php,asp,asp.net,vb,vbscript,javascript,c#,RSS

All content ©CommunityMX 2002-2004. All rights reserved.