

You are here: [Home](#) » **Content Query Web Part Alternative for WSS 3.0**



While working for a small client, I was tasked to create a web part for displaying list items from separate site collections/sites. Typically, in MOSS, I would have chosen the Content Query Web Part, but because of the constrains: (1) client is using WSS 3.0 and (2) the site is hosted through a shared-hosting service provider – the CQWP or other custom development solution was not an option.

First let me say that I'm sure this isn't the only way to accomplish this. Getting a little creative, I thought of other ways to simply retrieve the content from other lists.

My solution? [XML Web Part](#)

Ok, here's my story:

1. Create a list or library that you want to access content from (if you already have the list, skip this step)
2. From the Actions menu, click on [View RSS Feed](#)

A new window should open with an XML view of the items in your list.

3. Copy the URL from the browser.

4. On the destination page, add a new [XML Web Part](#) and choose Edit > [Modify Shared Web Part](#)

5. In the [XML Link](#) text box paste the link to the RSS feed for your list. You can shorten up the URL by removing the fully-qualified domain name:

e.g. http://www.site.com/web/_layouts/listfeed.aspx?List=<guidhere>

to

/web/_layouts/listfeed.aspx?List=<guidhere>

6. Click on the [XSL Editor](#) button and paste the following XSL stylesheet code into the text box.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/rss">
<html>
<head>
<link rel="stylesheet" type="text/css" href="/_layouts/1033/styles/core.css?rev=5msmprmeONfN6J3wtbAIA%3D%3D" mce_href="/_layouts/1033/styles/core.css?rev=5msmprmeONfN6J3wtbAIA%3D%3D"/>
<link href="xsl.css" mce_href="xsl.css" rel="stylesheet" type="text/css" />
<style type="text/css"> body { font-size:0.83em; } </style>
</head>
<body>
<div class="Snippet" style="border-width:0; background-color:#FFF; margin:0em">
<dl>
<xsl:for-each select="channel/item">
<dd><li style="list-style: square inside; color:gray;">
<xsl:element name="a">
<xsl:attribute name="href">
<xsl:value-of select="link"/>
</xsl:attribute>
<xsl:value-of select="title"/>
</xsl:element>
</li></dd>
<dt>
<xsl:value-of select="description" />
</dt>
</xsl:for-each>
</dl>
</div>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

You can customize this to your liking. It is pretty basic to start, but it does the trick.

7. Click OK

8. Make any other customizations to the web part, and click OK to apply the changes.

Done!

Like I said, it's ugly, but it works.

If anyone has any other suggestions on how to accomplish this, or a better stylesheet to use – please comment or message me! I am definitely NOT an XSL guy. 😊

Filed under: Uncategorized

Your email is never shared.
Required fields are marked *

Name *

Email *

Website

Allowed tags: <abbr title=""> <acronym title=""> <blockquote cite=""> <cite> <code> <del datetime=""> <i> <q cite=""> <strike>

• September

• 19

• 2007

While working for a small client, I was tasked to create a web part for displaying list items from separate site collections/sites. Typically, in MOSS, I would have chosen the Content Query Web Part, but because of the constraints: (1) client is using WSS 3.0 and (2) the site is hosted through a shared-hosting service provider – the CQWP or other custom development solution was not an option.

First let me say that I'm sure this isn't the only way to accomplish this. Getting a little creative, I thought of other ways to simply retrieve the content from other lists.

My solution? **XML Web Part**

Ok, here's my story:

1. Create a list or library that you want to access content from (if you already have the list, skip this step)

2. From the Actions menu, click on [View RSS Feed](#)

A new window should open with an XML view of the items in your list.

3. Copy the URL from the browser.

4. On the destination page, add a new [XML Web Part](#) and choose Edit > [Modify Shared Web Part](#)

5. In the [XML Link](#) text box paste the link to the RSS feed for your list. You can shorten up the URL by removing the fully-qualified domain name:

e.g. http://www.site.com/web/_layouts/listfeed.aspx?List=<guidhere>

to

/web/_layouts/listfeed.aspx?List=<guidhere>

6. Click on the [XSL Editor](#) button and paste the following XSL stylesheet code into the text box.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/rss">
<html>
<head>
<link rel="stylesheet" type="text/css" href="/_layouts/1033/styles/core.css?rev=5msmprmeONfN6J3wtbAIA%3D%3D" mce_href="/_layouts/1033/styles/core.css?
rev=5msmprmeONfN6J3wtbAIA%3D%3D"/>
<link href="xsl.css" mce_href="xsl.css" rel="stylesheet" type="text/css" />
<style type="text/css"> body { font-size:0.83em;} </style>
</head>
<body>
<div class="Snippet" style="border-width:0; background-color:#FFF; margin:0em">
<dl>
<dl>
<xsl:for-each select="channel/item">
<dd><li style="list-style: square inside; color:gray;">
<xsl:element name="a">
<xsl:attribute name="href">
<xsl:value-of select="link"/>
</xsl:attribute>
<xsl:value-of select="title"/>
</xsl:element>
</li></dd>
<dt>
<xsl:value-of select="description" />
</dt>
</xsl:for-each>
</dl>
</div>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

You can customize this to your liking. It is pretty basic to start, but it does the trick.

7. Click OK

8. Make any other customizations to the web part, and click OK to apply the changes.

Done!

Like I said, it's ugly, but it works.

If anyone has any other suggestions on how to accomplish this, or a better stylesheet to use – please comment or message me! I am definitely NOT an XSL guy. 😊

Filed under: Uncategorized

Your email is never shared.
Required fields are marked *

Name *

Email *

Website

Allowed tags: <abbr title=""> <acronym title=""> <blockquote cite=""> <cite> <code> <del datetime=""> <i> <q cite=""> <strike>

Copyright ©2009 Eric Kraus SharePoint Blog

•