# WorleyParsons SharePoint Development

by Jonny Trees

## Developing SharePoint Web Parts

As a SharePoint developer, you should investigate the out of the box SharePoint web parts and their capabilities to be extended, before thinking of rolling your own. MOSS provides many web parts and this is the basis of your toolset.

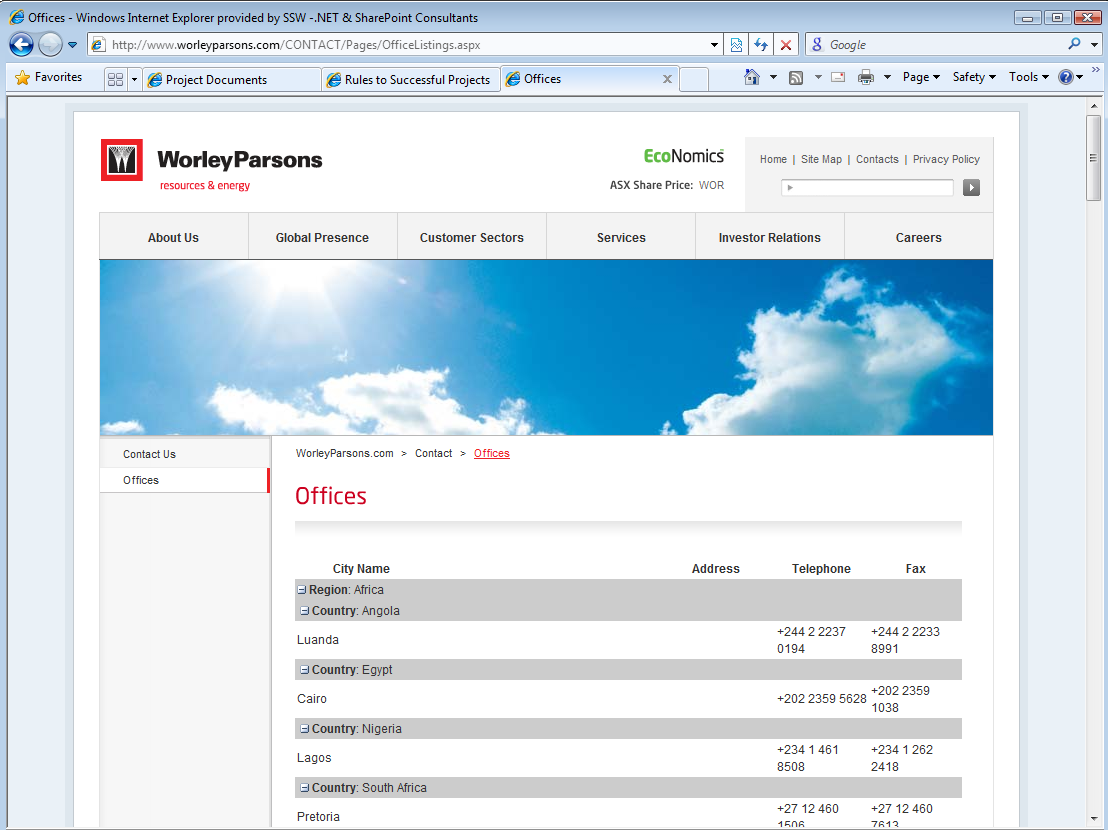
## Requirements

WorleyParsons required many web parts on their site. Let’s take a look at two of their requirements, see which web parts to choose, and what customizations to make.

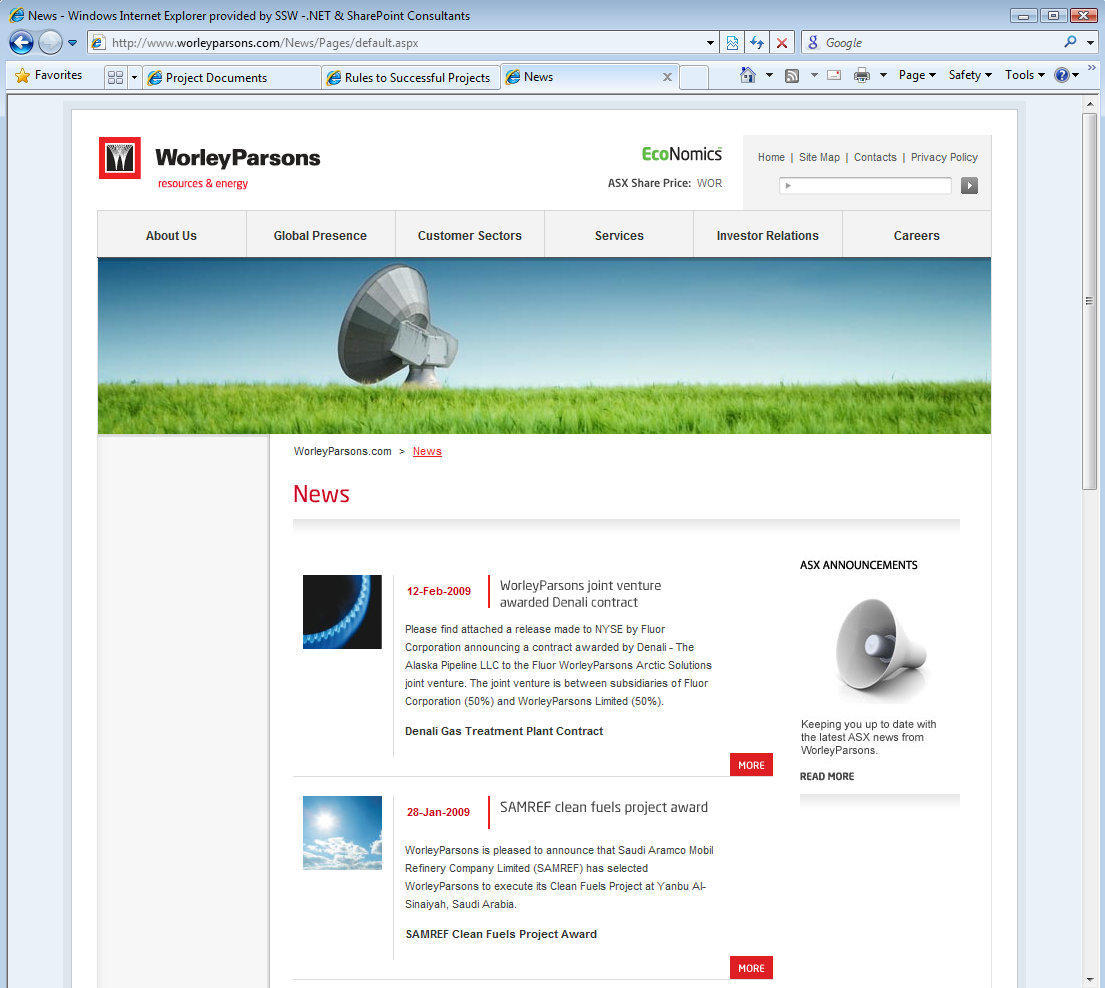
1. Office Listings web part
   * A list of all WorleyParsons offices and their related contact details
   * Grouping of offices by region and country
   * Anchor tags to allow direct navigation to the individual office, the region or the country e.g. /OfficeListings.aspx#NewZealand
   * The data should be easily modifiable
2. News Highlights web part
   * A list of several news items in summary form, ordered by the release date
   * A link provided to each full news article
   * News data pulled from individual sites or the whole site collection
   * The data should be easily modifiable

## Design Required

The above requirements were translated into this look and feel via a Photoshop images:



**Figure: The Office Listing web part that was required**



**Figure: The News Highlights web part**

## Office Listings – Getting the Data View Looking Right

The Office Listings was implemented using a Data View, customized through SharePoint Designer. There are several benefits to using the data view including:

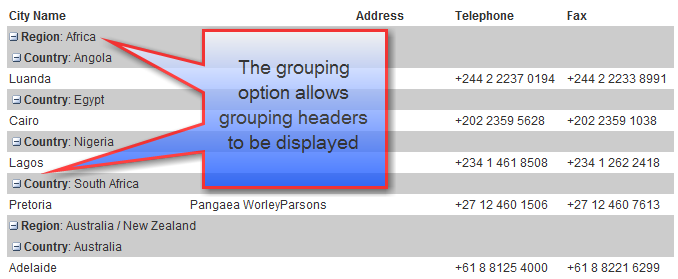
* Data source options such as SharePoint lists, web services, databases or XML
* Filtering, grouping and paging of data
* Viewing of individual items or multiple items in table format

**Part 1 – Creating the List**

The office listings consisted of about 200 records. We decided to store the office listings information in a SharePoint list. This provides easy editing by the content editors and is the least work for a developer. Beyond the standard fields such as address and phone number, we included a Boolean “Show” field that allowed content editors to toggle whether a specific offices contact details were displayed.

**Part 2 – Getting the Grouping Right**

In the Data View we took advantage of the grouping and filtering options. A filter was applied to only show records that had “Show” set to true, and grouping was applied to the Region and Country options to create the following effect:



**Figure: There are two levels of grouping – Region and Country**

**Part 3 – Getting the Anchor Tags Right**

The final requirement that had to be satisfied was the addition of anchor tags to regions, countries and individual offices. The Data View web part embeds the XSL style sheets into the web part so this could be directly manipulated to create anchor tags around each element.

To apply this to the two group headers (region and country), find the following text for each group header tag e.g.

<xsl:template name="dvt\_1.groupheader0">

Then add the following code within the first <td> tag of the group header:

<a>

<xsl:attribute name="name">

<xsl:value-of select="translate(normalize-space($alinkRegion),

' ','')" />

</xsl:attribute>

</a>

XSLT functions “translate” and “normalize-space” were used to manipulate the names of the offices, countries and regions.

normalize-space($alinkRegion)

Normalize-space removes leading and trailing white space and will compact all sequences of white space characters into a single character

e.g. “ Australia ” transformed to “Australia”

translate(*argument1*)

The Translate function replaces your instances of the first argument with the second argument, so in this case it’s used to remove any white space characters.

Tip: Although you would expect “translate” to remove all spaces, a combination of both functions was required to guarantee removal of all white space.

e.g. “Australia / New Zealand” transformed to “Australia/NewZealand”

The use of these functions allowed regions with a space in the name e.g. “Australia / New Zealand” to be compacted into “Australia/NewZealand”. This will therefore work correctly when used as the anchor tag name.

Similar code was used to add anchor tags to both the country and the office name.

## News Highlights – Styling the Content Query Web Part

To implement the News Highlights web part, we decided to customize the Content Query Web Part (CQWP). This web part allows content across a site collection or specific site to be aggregated and displayed in summary form.

Out of the box, the CQWP provided us with the following functionality:

* Content filtering by site (this allows the CQWP to show a summary of news articles from the Australian or African site for example)
* Content filtering by content type (we used a news content type for all news articles so we could ensure only content of this type would appear)
* Sorting of data (we used this to filter by the news article creation date)
* Item limit (the number of items displayed in the CQWP can be limited to 3 for example)

The CQWP provides several default styles for displaying content that use standard site columns such as Title and Description. We required a custom style that included site columns created for our News content type, and a customized look and feel.

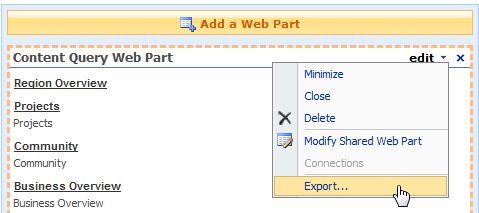
We took the following steps to build the required web part:

**Step 1 – Creating the Data Store**

A content type and a page layout were creating to define and create news pages. The content type included fields such as a News Abstract and a news Rollup Image. New pages could then be added to a site and serve as the data store for the New Highlights web part.

**Step 2 – Export web part**

After a Content Query Web Part is dropped into a Web Part Zone, it can be exported into its XML definition through the Edit, Export option:

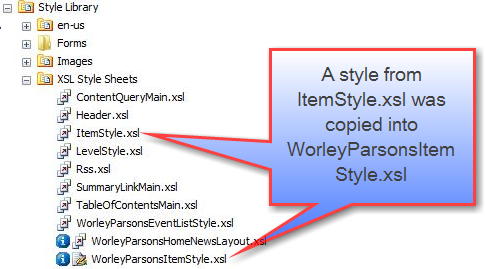


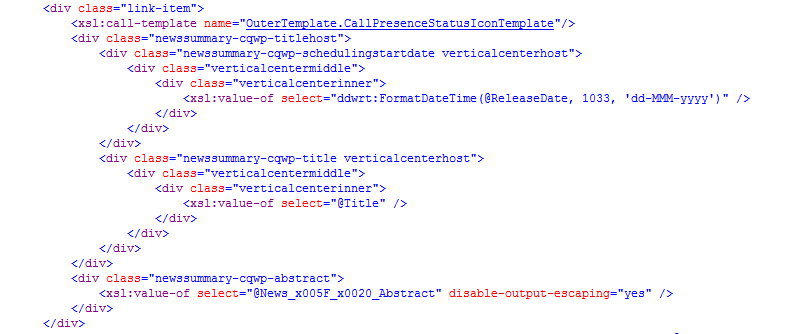
**Figure: Export the web part so you can customize the XML to add additional fields, apply a custom XSL style sheet etc.**  
  
The exported .webpart file can be modified in a text editor.  
  
Tip: if you configure the vanilla CQWP as much as possible through the SharePoint site, these settings are exported with the web part and will be set when this web part is redeployed

**Step 3 – Add the custom fields to the display of the CQWP**  
Properties in the XML can be modified to further customize your CQWP. To gain access to additional fields on your content type, you need to specify the site column names in the CommonViewFields property:  
  
 <property name="CommonViewFields" type="string">Rollup\_x0020\_Image,Image;News\_x0020\_Abstract,RichHTML;ReleaseDate,DateTime</property>

Note: “\_x0020\_” is used to replace the space character if spaces are used in site column names (see example above)

This is a semi-colon delimited list of site columns and their data types. Note that there should be no semi-colon at the end of the list.  
  
To determine the site column names, and for a list of site column data types, see Heather Solomon’s excellent and informative series of articles at: <http://www.heathersolomon.com/blog/articles/CustomItemStyle.aspx>   
  
Tip: When creating site columns, do not include spaces in the site column names. Immediately after creating the site column you can rename the site column to include spaces, this means that the underlying site column will not include spaces, but the site column name used in interfaces across your SharePoint site will have a user friendly name.

**Step 4 – Creation of custom style**  
  
The CQWP uses XSL style sheets for styling. To style the News Highlights web part, a style from the SharePoint ItemStyle.xsl file was copied into a new file:  
  


**Figure: Working with the XSL files in SharePoint Designer**  
  
This style was then modified to provide the styling we needed. This included references to CSS styles contained in the master CSS file.  
  


**Figure: The modified XSL style for the News Highlights Web Part**

Note: in the XSL styles, “\_x005F\_x0020\_” is used to replace the space character.  
  
Tip: the DDWRT script can be used as in the example above for formatting dates in XSL. Simply include the following line in the “xsl:stylesheet” tag:  
  
xmlns:ddwrt="http://schemas.microsoft.com/WebParts/v2/DataView/runtime”  
  
Finally the new style must be referenced in the web part in both the XSL property, and the ItemXslLink property.

Deployment  
  
The web part can now be added back into SharePoint. By navigating to the site collection Web Part Gallery, the web part can be uploaded and will then be available to drop into Web Part Zones on individual pages:

Some considerations when using the CQWP include the lack of paging, and the limit of querying 1000 lists at a time.

## Summary

We leveraged the out of the box SharePoint web parts on the WorleyParsons public website for displaying the news summary and the office listings information. The Content Query Web Part and the Data View control are powerful and highly extensible controls that produced excellent results.

Jonny Trees

<http://www.ssw.com.au/SSW/Employees/employeesprofile.aspx?EmpID=JRT>