To create a language selector on a page, use the Language data source web part in combination with the Basic repeater web part.

1. Place the **Language data source** web part on a page. The data source generates 'CultureCode', 'CultureName', and 'URL' (the URL is culture specific) fields.
2. Configure the web part and copy the value of the **Web part control ID** property.
3. Place the **Basic repeater** web part on the page and configure it.
4. In the **Data source name** property, paste the copied **Web part control ID** value from the data source web part.
5. In the **Content before** and **Content after** field, add the markup for the language selector. For example:

Content before

```
<div class="language-selection">
  {% LocalizationContext.CurrentCulture.CultureShortName %}
</div>
<nav role="navigation">
  <ul class="language-items">
```

Content after

```
  </ul>
</nav>
```

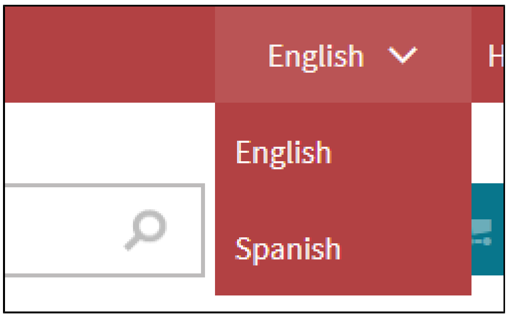
6. Access the data source fields in the transformation. For example:

Text / XML transformation code

```
<li>
   <a title="{% CultureName %}" href="{% HTMLEncode(URL) %} ">{% CultureName %}<
/a>
</li>
```

7. **Save & Close**.
8. Style the markup in the **CSS stylesheets** application.

The page now lists the available cultures.



Localizing the names of cultures

By default, the system displays the names of cultures in English. You can create resource strings for specific culture codes and access them using the following macro code:

Text / XML transformation code

```
{%
  cultureLocalizedName = GetResourceString(CurrentSite.SiteName + "." + CultureCode);
  return;
%}
```

You would then use the *cultureLocalizedName* variable in the transformation code instead of the *CultureName* source field. This allows you to display custom culture names instead of culture codes. The **Dancing Goat** sample site uses the same approach to build its language selector.

Renaming the language selection parameter

The default name of the language selection parameter is *lang*. To rename the parameter:

1. Add the **CMSLanguageParameterName** key into the */configuration/appSettings* section of your application's **web.config** file.
2. Enter the new name as the key's **value**.

```
<add key="CMSLanguageParameterName" value="sprache" />
```

Even though switching between languages using URL query string parameters is the default approach, it is not recommended for live production websites. Parameters make URLs more confusing and are not friendly toward search engines. Instead, you can configure the system to use one of the options described in Configuring URLs for multilingual websites.