

# Exercise 1 - Changing vision identity (banner)

*The banner of your GeoNetwork site can be easily changed by replacing the default graphics with graphics of your own organization. Objective of this exercise is illustrating the steps to update site's vision identity.*

To conduct the exercise you will need to perform the following steps:

1. design or select the graphics for the banner
2. copy the graphic files in the required folder
3. update the XSL file generating the banner (*banner.xsl*) and the CSS document.
4. refresh the home page

## The steps in more detail

**Step 1.** For this exercise, we borrowed graphic files from the Commonwealth Office of Spatial Data Management banner:

*my-banner-left.gif*  
*my-banner-back.gif*  
*my-banner-right.gif*

that are located in */exercises/ex02/* folder. In real life, you would design your own banner and create three parts like below.

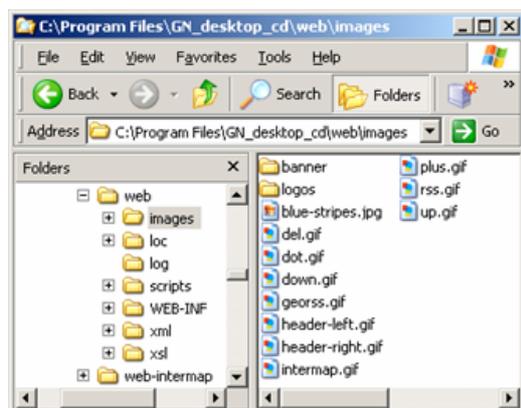


**Step 2.** The default banner consists of three graphics: a left part, a center part and a right part.

The **center** part is defined as a background image of the table the whole banner is assembled in. It is a graphic that can easily be repeated horizontally in a seamless way, linking the left and right graphic. The default graphic is named *blue-stripes.jpg*.

The **left** and **right** part are named *header-left.gif* and *header-right.gif* respectively. All banner graphics are 80 pixels high.

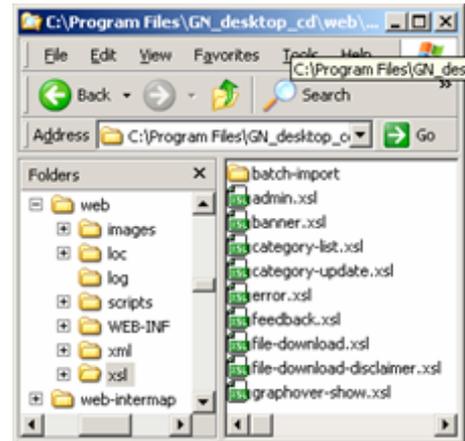
You can replace the default graphics with graphics of your own organization, but for now use the sample ones provided with this exercise. Copy the three files of the new banner in the *web/images/banner/* folder.



**Step 3** Open the file *banner.xsl* in a text editor or an XML editor. You will find this file in the *web\xsl/* folder.

The XSL files are responsible for the generation of the HTML output of GeoNetwork opensource. They operate on an incoming XML document, taking content from that XML document and transforming it into HTML output a web browser can display (render).

You will see the text listed below at the top of the document.



```
<xsl:template name="banner">
<table width="100%">
  <!-- title -->
  <tr class="banner">
    <td class="banner">
      
    </td>
    <td align="right" class="banner">
      
    </td>
  </tr>
  <!-- buttons -->
  <tr class="banner">
```

Note the highlighted text **in red** that defines the names and location of the left and right graphics. Note the highlighted text **in green** that defines the style used to render the middle part of the banner.

The `{root/gui/url}` will be replaced with the base URL of your GeoNetwork opensource installation; in this case. The rest of the string will point the browser to the appropriate left or right graphic.

**Replace the strings in red (above) with the ones below to match the new files.**

```
{root/gui/url}/images/banner/my-banner-left.gif
```

```
{root/gui/url}/images/banner/my-banner-right.gif
```

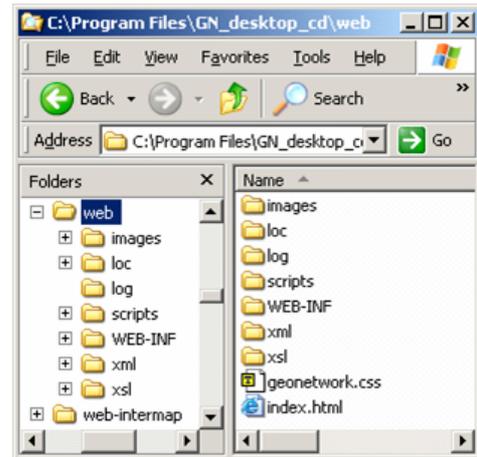
If you wish, change the *alt* value with more appropriate information.

**Save and close the document.**

Now **open** *geonetwork.css* file in a text or CSS editor. The file is located in the `\web` folder of your installation. We will have to update the image name of the graphic used for the middle part of the banner.

Search for the `td.banner` style in this document and update the filename accordingly (see below). Then, save and close the style sheet.

```
td.banner {  
    font-size: 8pt;  
    color: #ffffff;  
    /* background-image: url('images/blue-stripes.jpg'); old */  
    background-image: url('images/banner/my-banner-back.gif');  
    border-bottom: 1px solid #ffffff;  
}
```



**Step 4** Open your web browser and refresh the GeoNetwork opensource homepage.

Below is the expected result of updating the banner. Check if your changes are consistent.



## Exercise 2 - Changing text

All text that is part of the GeoNetwork opensource user interface is maintained in a set of XML files. These files can be localized to provide access to the site in several languages. The exercise shows how to insert a new text string in the home page.

To conduct the exercise you will need to perform the following steps:

1. open strings XML files and add a new element
2. open *main-page.xsl* and add a new XSL element
3. save updates and refresh the home page

### The steps in more detail

Before proceeding with the exercise, some notes about localization.

To see the localized files, open your file manager and go to location `\web\loc\`.

For each available language, you will find a two letter coded folder (*en*, *es*, *fr*) containing an *images* and an *xml* folders.

Open the ***strings.xml*** file (*en\xml\*) in a text or XML editor. *Strings.xml* contains most of the generic user interface text enclosed in user tags.

The structure of the files is relatively simple. The file starts with a **root element**, followed by elements containing strings. Elements can also contain HTML code or can contain attributes.

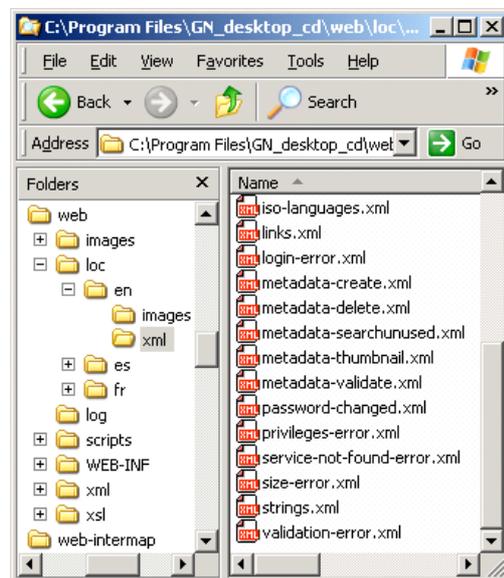
Each element has a logical name, e.g. `<admin>`, and contains the string to appear in the user interface. For example the *mainpageTitle* element (`<mainpageTitle>...</mainpageTitle>`) contains the title of the search box in the home page.

**Use the search tool and search for *mainpageTitle* element to see its content. Also check out the same element in different languages.**

Following is the XSL element that allows the transformation in HTML string of the tag value.

```
<xsl:value-of select="/root/gui/strings/mainpageTitle"/>
```

This element is used in *main-page.xsl*. The figure in the next page shows the output.





Logically, the `en/xml` folder should contain XML files with English strings, while the `fr` folder should contain the same XML files, but now containing French strings and so on. Note that the structure of the XML file is very simple.

A nice feature of GeoNetwork opensource is that it has a default language, **set to English**. When viewing the website in another language, all strings that are not translated will be displayed in the default language. This ensures that no text disappears from the GUI.

**Step 1.** The exercise will now take you through the steps of adding a new localized string at the bottom of the search box of `main-page.xsl` like you see in the figure on the right.



Open `strings.xml` and add a new element as showed below. Insert the element at the bottom of the file just above the root element (`</strings>`).

```
<workshopText>FOSS4G Conference, Lausanne, 12-15 September 2006</workshopText>
```

**Do it for each language** (`en`, `es` and `fr`) and translate the text (translations are in `exercises/ex04/strings.txt` file).

**Step 2.** Open `main-page.xsl` and find the commented section "Info" in the content template (see red box on the right).

```
<tr height="100%">
  <!-- Info -->
  <td valign="top">
    <xsl:copy-of select="/root/gui/strings/mainpage1"/>
    <xsl:copy-of select="/root/gui/strings/mainpage2"/>
  </td>
</tr>
```

Add the next lines after the code above.

```
<!-- additional text for training -->
<tr><td height="30px"></td></tr>
<tr>
  <td align="right">
    <h3><xsl:value-of select="/root/gui/strings/workshopText" /></h3>
  </td>
</tr>
<!-- END additional text for training -->
```

**Step 3.** Save `main-page.xsl` and refresh the browser window.

The home page should now show the added text at the bottom of the search box. Try different languages to see how the localization works.

# Exercise 03 – Creating an XML export service for metadata

*Creation of a simple new service on GeoNetwork opensource that will export a metadata record in XML format. The service will use existing functionality from the GeoNetwork engine, so no Java programming is required. Note: For simplicity, this exercise will focus on ISO19115 formatted metadata only.*

## 1) Pre-configure the server

- Open WEB-INF/config.xml
- Locate <debug>>false</debug> and set to true
- Restart GeoNetwork
- Type

http://127.0.0.1:8080/geonetwork/srv/en/metadata.show!?id=15&currTab=simple (notice the "!" after the service name)

- Notice that the full metadata XML is directly inside of the <root> element. We'll use this to write the XPath expression to catch the metadata part from the XML output.

## 2) Create a new service

GeoNetwork services are based on the Jeeves framework. These are defined in the WEB-INF/config.xml configuration file. We start by copying an existing service and then we'll modify it as needed:

- Open the file WEB-INF/config.xml
- Locate the service metadata.show:

```
<service name="metadata.show">
  <class name=".services.metadata.Show" />
  <output sheet="metadata-show.xsl">
    <call name="services" class="jeeves.guiservices.services.Get" />
    <call name="currTab" class=".guiservices.util.GetCurrentMDTab" />
    <xml name="isoLang" file="xml/iso-languages.xml" />
    <xml name="sources" file="xml/sources.xml" localized="false" />
  </output>
  <error id="access" sheet="error-form.xsl">
    <xml name="privError" file="xml/privileges-error.xml" />
  </error>
</service>
```

- Copy the block and paste it to the end of file, before "</service>", and change as follows (changes are highlighted in bold):

```
...
<service name="metadata.getAsText">
  <class name=".services.metadata.Show" />
  <output sheet="metadata-show.xsl">
    <call name="services" class="jeeves.guiservices.services.Get" />
    <call name="currTab" class=".guiservices.util.GetCurrentMDTab" />
    <xml name="isoLang" file="xml/iso-languages.xml" />
    <xml name="sources" file="xml/sources.xml" localized="false" />
  </output>
  <error id="access" sheet="error-form.xsl">
```

```

        <xml name="privError" file="xml/privileges-error.xml" />
    </error>
</service>
</services>
<!--
===== -->
</geonet>

```

### 3) Set privileges for the new service

Now we have to set the privileges from the service we just created. Services are defined in the file `xml/user-profiles.xml`:

- Open `xml/user-profiles.xml`
- Add the following two line before "`</profile>`"

```

...
    <!-- Get metadata xml as plain text -->
    <allow service="metadata.getAsText"/>

    </profile>

    <!--
===== -->
</profiles>

```

- Restart geonetwork
- Try the `metadata.getAsText` service:

`http://127.0.0.1:8080/geonetwork/srv/en/metadata.getAsText?id=15`

- Not surprisingly the page is the same as `metadata.show...`

### 4) Customize the service output

Finally we modify the service definition to return XML as text/plain:

- Duplicate the stylesheet `potal-full.xsl` and rename it `metadata-get-as-text.xsl`
- Edit the file `potal-full.xsl` as follows (changes are highlighted in bold):

```

<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

    <xsl:template match="/root">
        <xsl:apply-templates select="/root/Metadata" />
    </xsl:template>

    <!-- default: just copy -->
    <xsl:template match="@*|node()">
        <xsl:copy>
            <xsl:apply-templates select="@*|node()[name() !=
'geonet:info']" />
        </xsl:copy>

```

