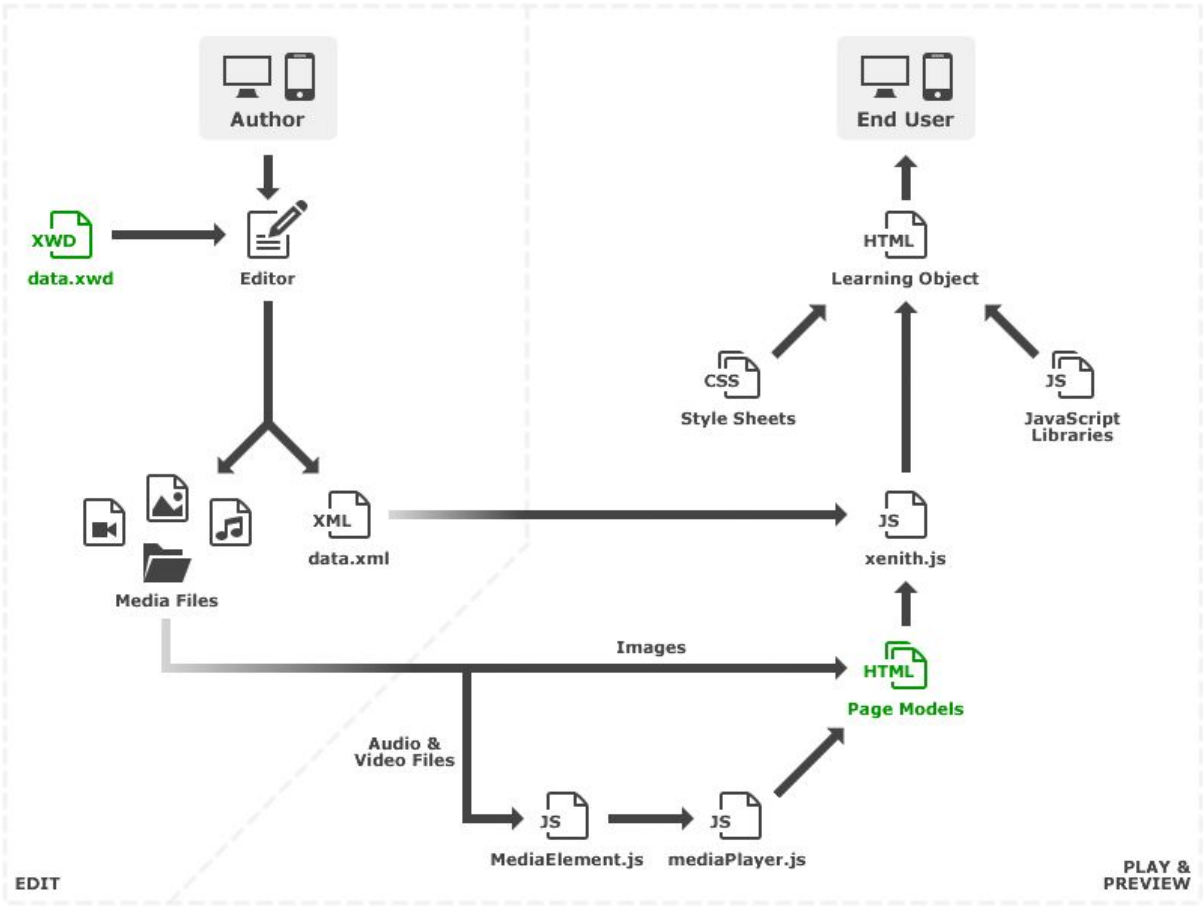
How to Edit an Existing Page Type

There are two files that can be edited to change the functionality of an existing page type. Let's look at the structure of the main files involved in editing and playing a Toolkits project:



The highlighted files show the two aspects of every page:

1. **XWD data** specifying how the page is created in the editor's wizard and saved to the XML file.
2. **HTML page model** specifying how the page appears in the interface to the end user.

To change a superficial aspect of a page, or something that will always affect pages of that type, you only need to edit the associated **page model** file.

If the change requires some extra information from the project's author to implement it, or if you want the author to be able to enable or disable the new functionality, then you will also need to edit the **data.xwd** file.

See 'The Structure of XOT Projects' document for more details on this structure.

Example: Making a Simple Change to a Page

This example will demonstrate how to change the Graphics and Sound page so that images are captioned. The caption used will be that given in the 'Image Description' field of the wizard, currently used as the alt text for accessibility. We will not change the XWD file as this field already exists and the author won't need to enter any additional information.

Step 1: Locate the Page Model

The pages models can be found in: modules/xerte/parent_templates/Nottingham/models_html5/…

Every page model has a unique name which matches the associated node for the page in the XWD file. Normally it is quite easy to work out which page model relates to each page type but if you're not sure you can refer to the XWD file.

Open modules/xerte/parent_templates/Nottingham/wizards/en-GB/data.xwd and search for the page name as it appears on the insert menu of the editor. For example, a search for 'Graphics and Sound' takes us to this node:

```
<textGraphics menu="Media" menuItem="Graphics and Sound" ... >
```

The page model name will match the node's name, so the page model we need to open is textGraphics.html.

Every page model consists of three parts: HTML elements, jQuery / JavaScript code & CSS styles. Any of these sections can be edited, as long as the general structure of the file isn't changed. See the 'How to Create a New Page Type' document for more details on the structure of page model files and what you should avoid changing.

Step 2: HTML

The existing HTML elements can be found towards the bottom of textGraphics.html.

First we will add a div tag so that the caption will appear below the image and audio player but within the image panel.

```
<div id="pageContents">

    <div id="imgHolder" class="mobileAlign">
        <div class="panel inline">
            <img id="pageImg" style="visibility: hidden" />
            <div id="pageAudio"></div>
            <div id="imgCaption"></div> <!-- new caption tag! -->
            <div id="transcriptHolder">
                <div id="transcript"></div>
                <button id="transcriptBtn"></button>
            </div>
        </div>
    </div>

    <div id="textHolder"></div>

</div>
```

You can see that we have given the div an id of 'imgCaption'. We will refer to this in the next two steps.

Step 3: Styles

Above the HTML elements you will see a style tag containing the existing CSS information.

The caption style we will add is shown below. It references an element with the id of 'imgCaption' and adds styles so that the caption text will be bold, aligned to the right and with padding between it and the image above.

```
<style type="text/css">

    #imgCaption {
        padding-top: 5px;
        font-weight: bold;
        text-align:  right;
    }

</style>
```

Step 4: JavaScript

Now we need to populate our imgCaption tag with the caption text. For this we will use jQuery to get the relevant information from the project's XML data and insert it into the tag.

The script tag containing the JavaScript code is at the top of textGraphics.html. It contains a function (named to match the page model name) within which are a number of further functions used to set up the page.

```
<script type="text/javascript">

    var textGraphics = new function() {

        this.pageChanged = function() {...}

        this.sizeChanged = function() {...}

        this.init = function() {...}
    }

    textGraphics.init();

</script>
```

We will insert our code inside the init function so that it appears as soon as the page is loaded into the interface, i.e. when it is first viewed.

```
$("#imgCaption").html(x_currentPageXML.getAttribute("tip"));
```

You can see in the code above that we select the element with an id of imgCaption and then set the HTML for inside this element to equal the value of the tip attribute from the current page's XML.

The xenith.js file loads each page into the interface as a project is navigated through. On each page change it sets the value of x_currentPageXML to contain the XML data for the currently shown page. The tip attribute will be the text that an author has entered into the 'Image Description' field in the editor.

See the 'How to Create a New Page Type' document for more detail on how you can reference XML data from within a page model.

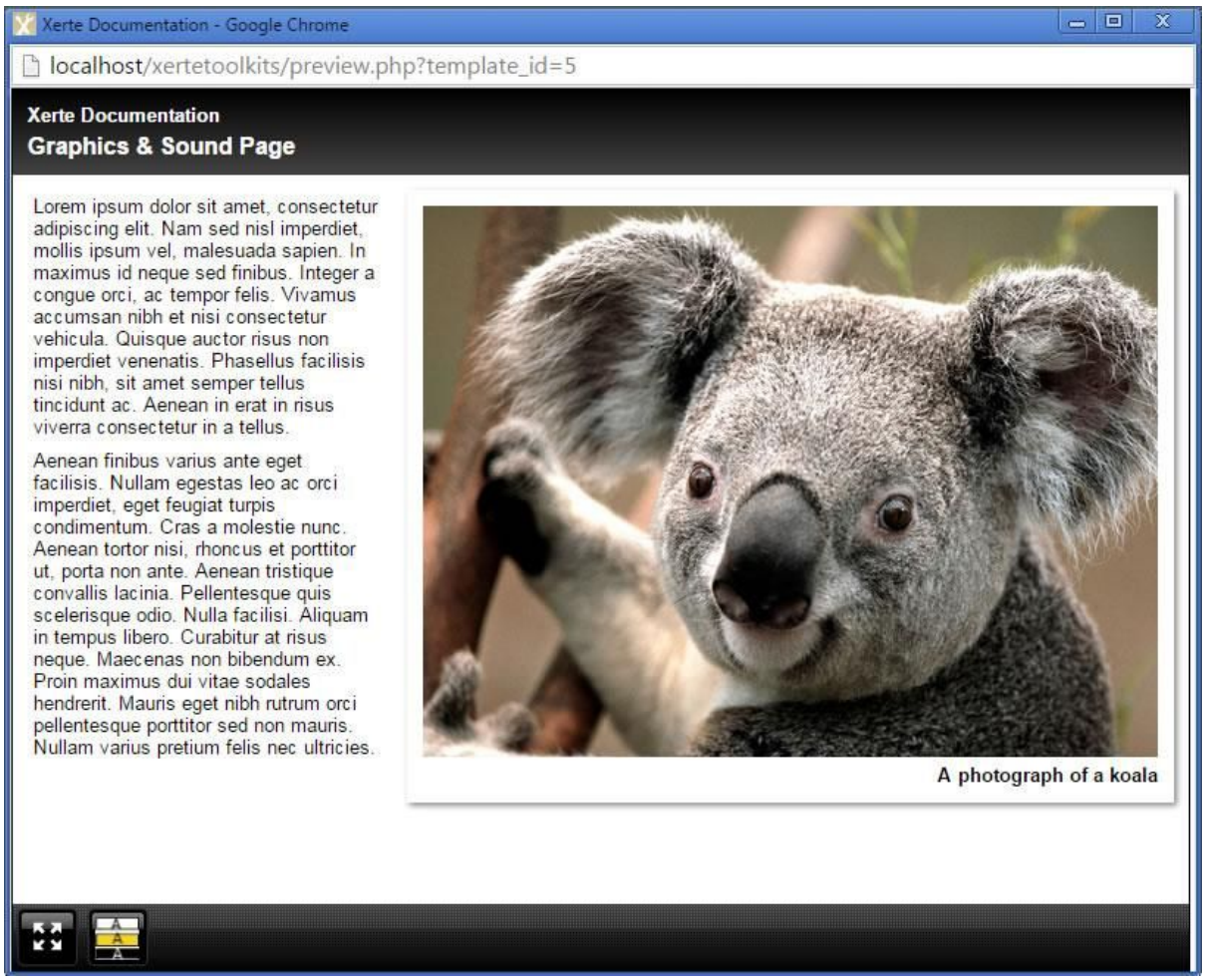
Step 5: Test

Once textGraphics.html has been saved you should test the new functionality works as you would expect.

Create a new project and insert a Graphics and Sound page.

Upload an image file and change the text in the 'Image Description' field to describe the image. Then press 'Play' to open the preview window.

You should see that the text entered as the Image Description appears as a caption below the image.



So, we have made a simple change to a page that takes some existing data (the image description) and uses it to add a new feature to the page (the caption). But what if we would like the author to have the option to not have a caption, or to have the alt text for the image (describing it for accessibility) to be different to the caption visible to everyone? To make these changes we will also need to edit the XWD data for the page to give the author more flexibility.

Example: Adding a New Option to the Wizard

This example will demonstrate how to continue changing the Graphics and Sound page so that the author has more control over the way images are captioned. We will add a new optional property that can be added to the page's XML via the editor, and it will be this that determines whether a caption is present.

Step 1: Locate the XWD File

The main XWD file is at modules/xerte/parent_templates/Nottingham/wizards/en-GB/data.xwd

However, we do not recommend editing the data.xwd files directly. If you make direct changes to data.xwd it will become time consuming to upgrade your installation without losing them.

The main data.xwd file is made from merging many smaller XWD files together. There is an individual XWD file for each page type, as well as one containing the global properties for a project (project title etc.).

These source files can be found at src/Nottingham/wizards/en-GB/ and they are named to match the associated page model. Therefore, the XWD file we will edit is textGraphics.xwd.

Step 2: Edit the XWD

The textGraphics.xwd file is structured as follows:

```
<wizard menus="Media">
  <pageWizard remove="true" duplicate="false">
    <newNodes>
      <textGraphics><![CDATA[<textGraphics name="My Name"/>]]></textGraphics>
    </newNodes>
  </pageWizard>

  <!-- TEXT GRAPHICS SOUND PAGE================================================
-->
  <textGraphics menu="Media" menuItem="Graphics and Sound" remove="true">

    <name label="Page Title" type="TextInput" wysiwyg="true" />
    <url label="Image" type="media" />
    <tip label="Image Description" type="TextInput" />

    <transcript label="Transcript" type="TextArea" height="100" optional="true"/>
    <sound label="Sound" type="media" optional="true"/>

  </textGraphics>
</wizard>
```

We will only make changes to the contents of the textGraphics element. This section of the XWD lists and describes the fields that will appear in the page's wizard.

You can see that textGraphics has five children (for ease of use this is a simplified version - there are more in the real XWD). The first three (name, url and tip) are always on the wizard for a page of this type as they are essential to the page working (they set the page title, image file and image alt text).

The next two children (transcript and sound) are optional properties as the author can choose whether to include this additional functionality. This is where we will add our caption option by including the following:

```
<caption label="Image Caption" type="TextInput" defaultValue="Enter a Caption for the Image" optional="true"/>
```

Our node has the name 'caption', which we will later refer to in the page model to retrieve the data the author enters in this field. It has a label (a caption to appear next to the field in the editor), a type (to set the type of field, e.g. text input or colour picker), a default value, and crucially it has the attribute 'optional' so that it will not be a default property applied to all pages of this type but will need to be manually added by the author.

See 'How to Create a New Page Type' document for more detailed information about the structure XWD files and how they relate to a project's XML.

Step 3: Rebuild data.xwd

Now we have edited the page's XWD file we need to rebuild the main data.xwd so that it includes our changes.

To do this you should run either build/rebuildNottingham.bat (Windows) or build/rebuildNottingham.sh (Linux).

Full details on how to rebuild data.xwd and debug any errors can be found at src/Nottingham/buildXWD.txt.

Step 4: Edit the Page Model

The editor is now set up to allow authors to add an optional caption to the page. We have also previously set up the page model to create a caption using the text entered in the Image Description field of the wizard. We will now edit the page model to use the new optional 'caption' property instead.

Open modules/xerte/parent_templates/Nottingham/models_html5/textGraphics.html and find the line of code we previously added:

```
$("#imgCaption").html(x_currentPageXML.getAttribute("tip"));
```

We no longer want the caption to use the 'tip' attribute to populate the caption text as we have set up a new one called 'caption'.

```
$("#imgCaption").html(x_currentPageXML.getAttribute("caption"));
```

But, because this property is optional, we can no longer assume it exists in the XML. We only want the caption div to appear if it has been added by the author and has a value of something other than a blank string.

```
if (x_currentPageXML.getAttribute("caption") != undefined &&
x_currentPageXML.getAttribute("caption") != "") {
    $("#imgCaption").html(x_currentPageXML.getAttribute("caption"));
} else {
    $("#imgCaption").remove();
}
```

Step 5: Test

Once textGraphics.html has been saved you should test the new functionality works as you would expect.

Create a new project, insert a Graphics and Sound page and upload an image file. Then press 'Play' to open the preview window.

You should not see a caption beneath the image.

Return to the editor, add the 'Caption' optional property to the wizard and complete the field with your caption text. When you preview the project again you should see that the text entered appears as a caption.