

SlideshowBox

The Source XML Structure

The source XML contains a list of *item* elements (an element is identified by a XML tag specified between the "<" and ">" characters), each *item* element (tag) having extra child elements that represent the URL for the images, text for the title and description and other extra (custom) parameters the user might require. So a typical source XML file might look like this:

```
<SlideshowBox>
  <items>
    <item>
      <thumbnailPath>URL for the thumbnail image 1</thumbnailPath>
      <largeImagePath>URL for large image 1 (mandatory)</largeImagePath>
      <fullScreenImagePath>URL for full screen image 1 </fullScreenImagePath>
      <title><![CDATA[The title for image 1 goes here]]></title>
      <description><![CDATA[Description for image 1 goes here]]></description>
    </item>
    <item>
      <thumbnailPath>URL for the thumbnail image 2</thumbnailPath>
      <largeImagePath>URL for large image 2 (mandatory)</largeImagePath>
      <fullScreenImagePath>URL for full screen image 2 </fullScreenImagePath>
      <title><![CDATA[The title for image 2 goes here]]></title>
      <description><![CDATA[Description for image 2 goes here]]></description>
    </item>
    <item>
      <thumbnailPath>URL for the thumbnail image 3</thumbnailPath>
      <largeImagePath>URL for large image 3 (mandatory)</largeImagePath>
      <fullScreenImagePath>URL for full screen image 3 </fullScreenImagePath>
      <title><![CDATA[The title for image 3 goes here]]></title>
      <description><![CDATA[Description for image 3 goes here]]></description>
    </item>
    . . .
    <item>
      <thumbnailPath>URL for the thumbnail image N</thumbnailPath>
      <largeImagePath>URL for large image N (mandatory)</largeImagePath>
      <fullScreenImagePath>URL for full screen image N </fullScreenImagePath>
      <title><![CDATA[The title for image N goes here]]></title>
      <description><![CDATA[Description for image N goes here]]></description>
    </item>
  </items>
</SlideshowBox>
```

The <item> element contains all the information regarding each image from the album.

Note: The text *written in italics* in the above example should be replaced by your actual image URLs, titles and descriptions.

<thumbnailPath>

This element contains the URL of the thumbnail images. If the album template uses thumbnails, this is where it will look for the path and file name of the thumbnails. If this value is missing, the album template will use the large images as thumbnails.

<largeImagePath>

This element is the most important one and it is **mandatory**. It contains the URL for the large images used in the normal screen mode. This value should **should not be missing**, otherwise the **albums will not work properly** or at all. The large images could also replace the thumbnails and the large images used in full screen mode.

<fullScreenImagePath>

This element contains the URL to the large images displayed in full screen mode. When entering in full screen, the album will load the large images specified by this element. If the value is missing, the albums will use the normal screen large images instead, specified by the <largeImagePath> element.

Note: The URL of the images and thumbnails can be specified either as absolute path – the full path containing the protocol (http://), domain name (www.mysite.com) and directory path (/slideshows/images/) - or the relative path to the images – relative to another location on the hosting server. If you are unsure of how to specify the path/URL to the images, we recommend using the absolute path, that is specifying the complete path/URL to the image to avoid confusion.

Example

- absolute path: *http://www.mydomain.com/slideshows/images/thumbnails/thumb01.jpg*
- relative path: *images/thumbnails/thumb01.jpg* – the path is relative to the slideshows folder on the root of the domain (*http://www.mydomain.com/slideshows/*).

<title> and <description>

These two elements contain the title and description for the images. The text in these two elements should be added as CDATA text.

Example

```
<![CDATA[This is a description of the image.]]>
```

Custom XML elements

Note: The custom XML elements of each item can only be accessed from within Adobe Flash applications (you need to use ActionScript 3.0 code to access them).

Besides the standard elements, you can also specify other elements representing extra properties for the image items in the album: extra descriptions for the images, prices associated

to the images (if the album would represent a list of products) or simply some other information you could use to execute different actions in your Flash application that contains the album. These extra elements will be then added as properties directly to the image item objects so you can then retrieve their values via ActionScript code: `myAlbum.imageList[2].title` or `myAlbum.imageList[4].myProperty`.

Example

```
<item>
  <largeImagePath>images/image4.jpg</largeImagePath>
  <link>http://www.somedomain.com/</link>
  <target>_blank</target>
  <companyName>Some Company, Inc.</companyName>
  <type>video</type>
  <videoURL>http://www.somedomain.com/videos/video1.flv</videoURL>
</item>
```

This way, by verifying the value of the element `<type>` (actually by checking the type property added to the item), you could load a video clip into a video player when that particular image is displayed.

Example

```
import com.jumpeye.events.SlideshowBoxEvent;

myAlbum.addEventListener(SlideshowBoxEvent.PHOTO_SHOW, checkForVideo);

function checkForVideo(e:SlideshowBoxEvent):void {
    var imageIndex:int = e.item.index;
    if (myAlbum.imageList[imageIndex].type == "video") {
        var videoURL:String = myAlbum.imageList[imageIndex].videoURL;
        loadVideo(videoURL);
    }
    else {
        showAlternativeContent();
    }
}
```