Getting Started with Open Exhibits Player

Whether you are an author/editor or a developer, the Open Exhibits Player is the best way to get started with Open Exhibits. Think of the Player as a browser that runs applications (we call them “exhibits”) written in Creative Markup Language (CML), Cascading Style Sheets (CSS) and Gesture Markup Language (GML). Once you know how to open and run example exhibits included with the Player you can create your own exhibit.

Level: Beginner

This tutorial is suitable for author/editors and developers who have never used Open Exhibits before. Authors/editors who modify and create applications CML and CSS without programming in ActionScript can continue working with the Player. Developers may want to move on to the advanced tutorials which explain how to use the Open Exhibits SDK.

What you should know already

Be familiar with installing software on your platform

What you will need

Access to install software on your computer, a touch-enabled screen is useful, but not required

How long will this take

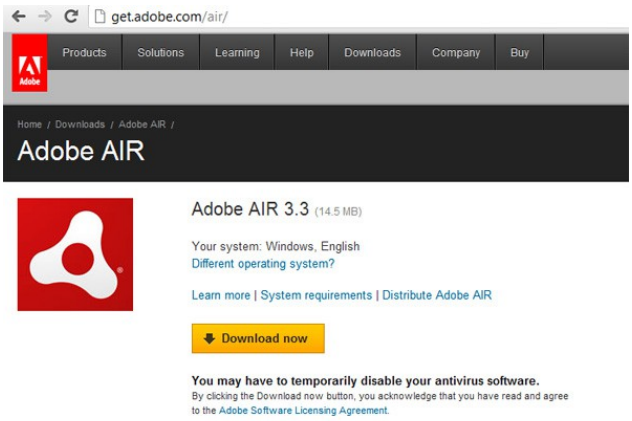
1 hour

In this tutorial you will

- Download and install the Player
- Open and run an example exhibit
- Test with a Touch Device or Simulator
- Customize and style the example exhibit

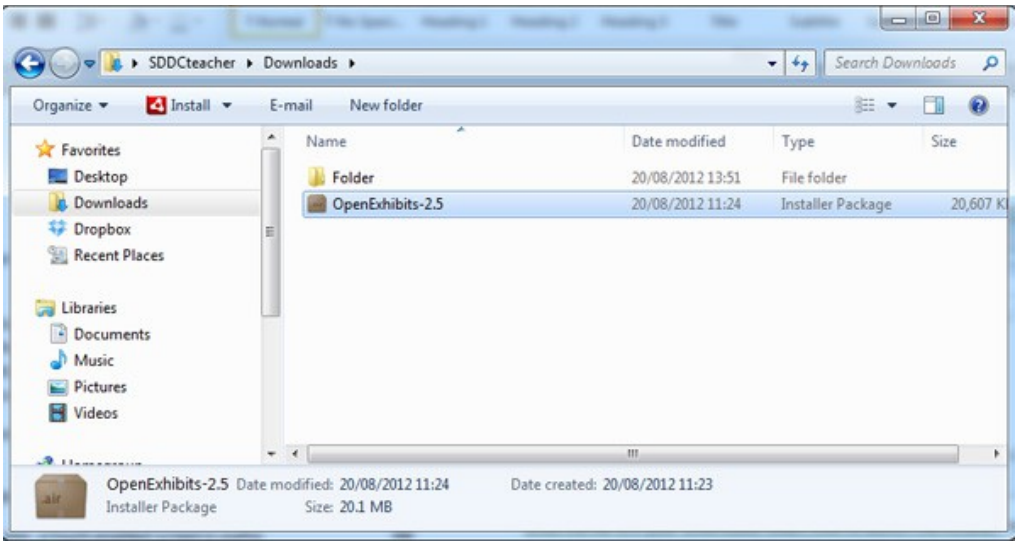
Step 1: Download and Install the Player

Check to see if you have Adobe AIR installed. This is the platform that both the Open Exhibits Player and Open Exhibits SDK require to run. If you do not have Adobe AIR, go to get.adobe.com/air/ and download the correct version for your computer. You will need administrator access on your computer to install it.



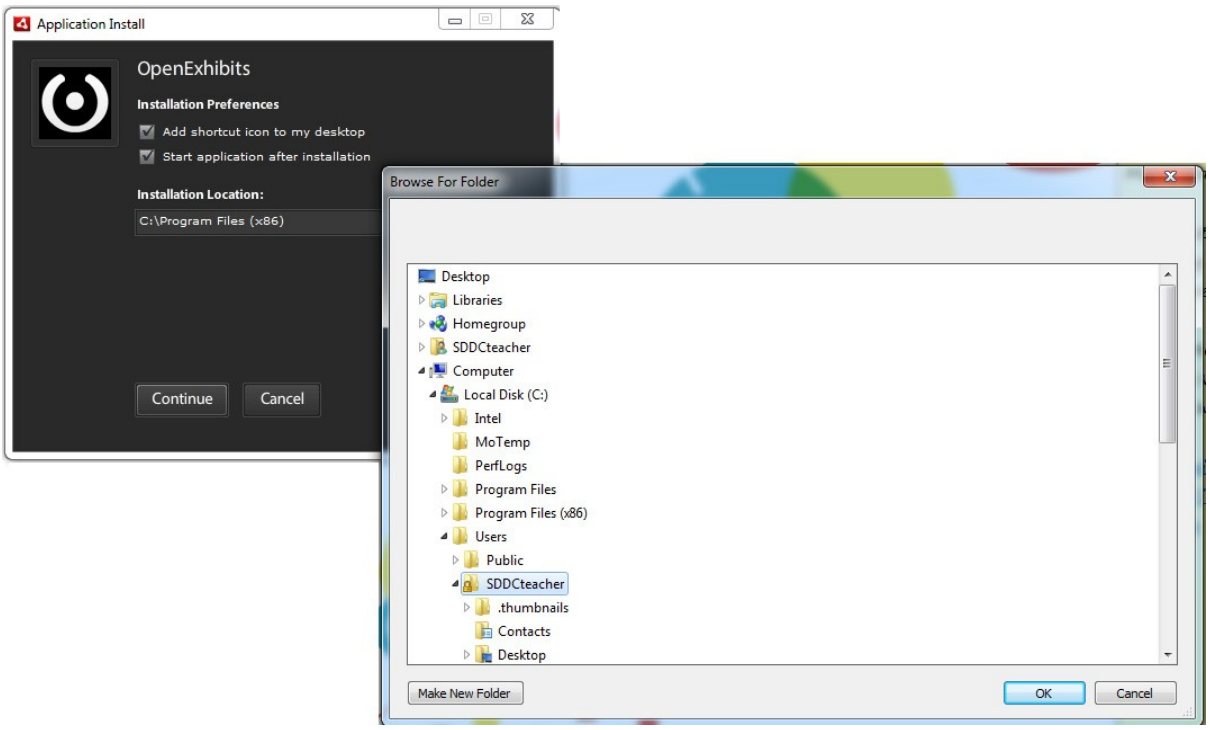
Once you have installed Adobe AIR, go to OpenExhibits.org. To download the Player, you must be a member of Open Exhibits, so create an account if don't already have one. Once you are logged in, find the Downloads section of the website.

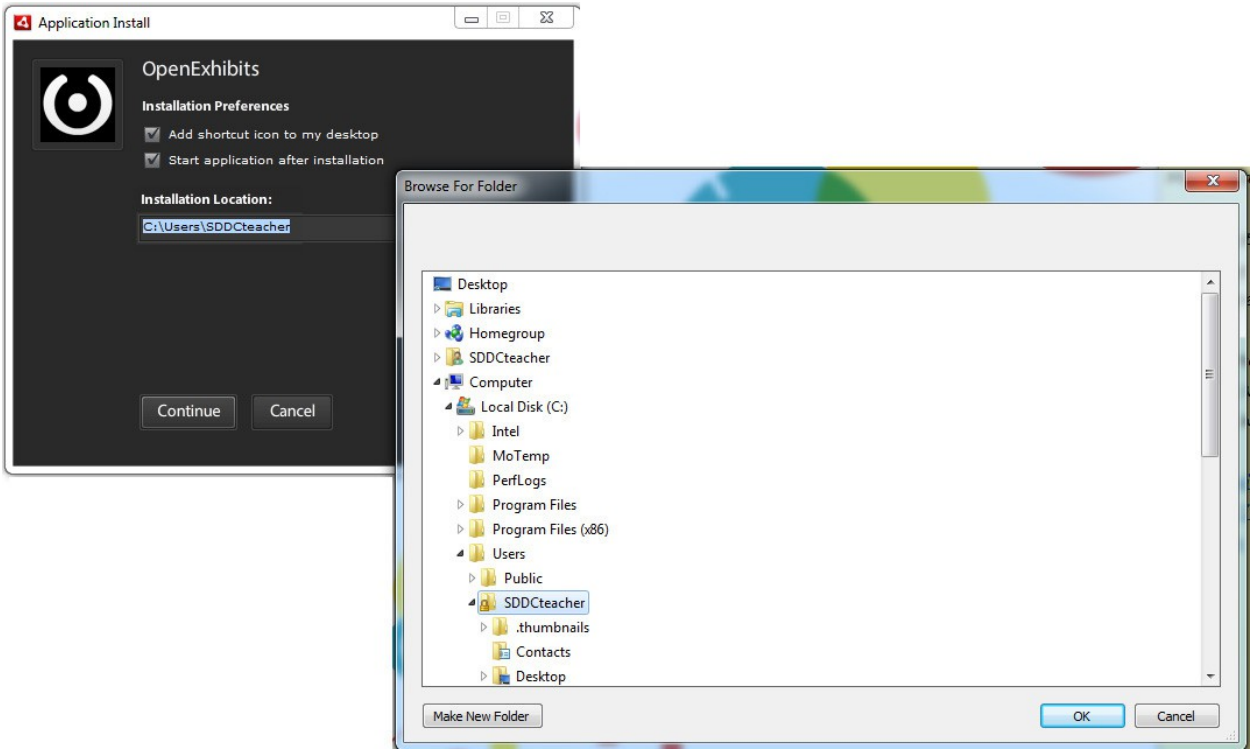
Select the Player download. The Player is an executable that allows you to open and run exhibits, including the examples included in the download. If you are an author/editor and plan to create exhibits by editing CML, then the Player is probably all you need. If you are a developer, you may want to download the Open Exhibits SDK after completing this tutorial. It contains additional libraries for programing in ActionScript.



Once you have successfully downloaded the Player (OpenExhibits.air), click to launch the installer. You will need administrator access on your computer to install it.

Launching the installer will bring up a dialogue box like the one above. Click Install to start the process.





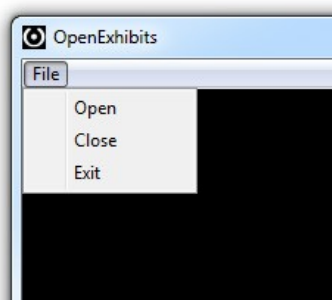
Next you will need to specify the install location. By default, the Player installs into your own User folder. This makes it possible for you to edit files easily without any permissions conflicts. We recommend you

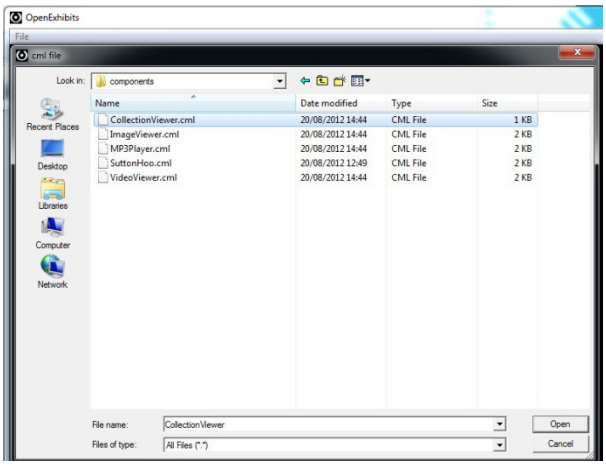
leave the default location, but if you prefer to install into another location choose Browse and then navigate to the folder you want to use.

When you have selected an install location, click Continue. You may need to give Adobe AIR permission to install on your computer. If a warning message appears, click Yes to allow the installation.

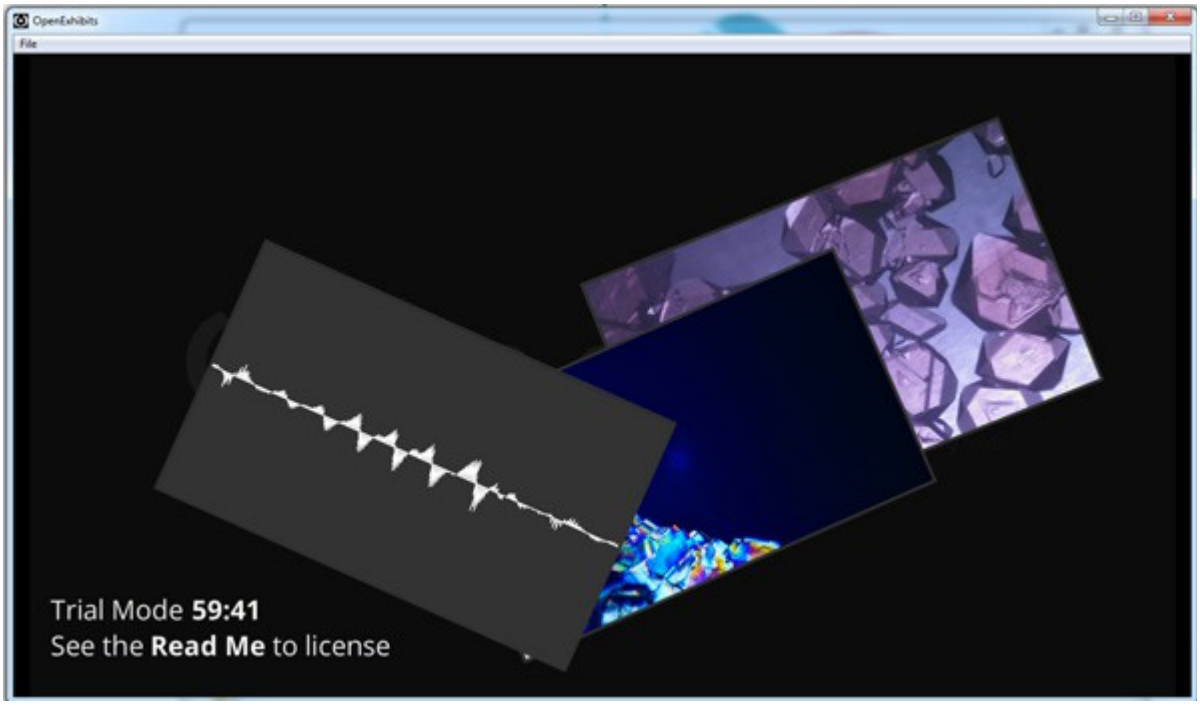
If you chose the default options during installation, then the Player will automatically launch in a new window when the installation is finished. A splash screen is displayed and there is a File menu at the top.

Step 2: Open and Run an Example Exhibit





Once you have the Player installed, you can run the examples included in the download. Click File > Open and then navigate to the Open Exhibits installation directory (probably "Computer/Users/YourUserName/OpenExhibits"). Go to the library/cml/components folder and select CollectionViewer.cml. You will notice other CML files in this folder. Each file corresponds to a "component," which in Open Exhibits is both an example for you to run, and a basic building block you can use to create your own exhibits.



The Collection Viewer is a very useful component. It is almost a mini-exhibit. The Collection Viewer allows you to browse and play three types of media: images, sounds and video. When the Viewer first opens the media elements are scattered randomly across the screen like a deck of cards. The user can drag, scale and rotate each element.

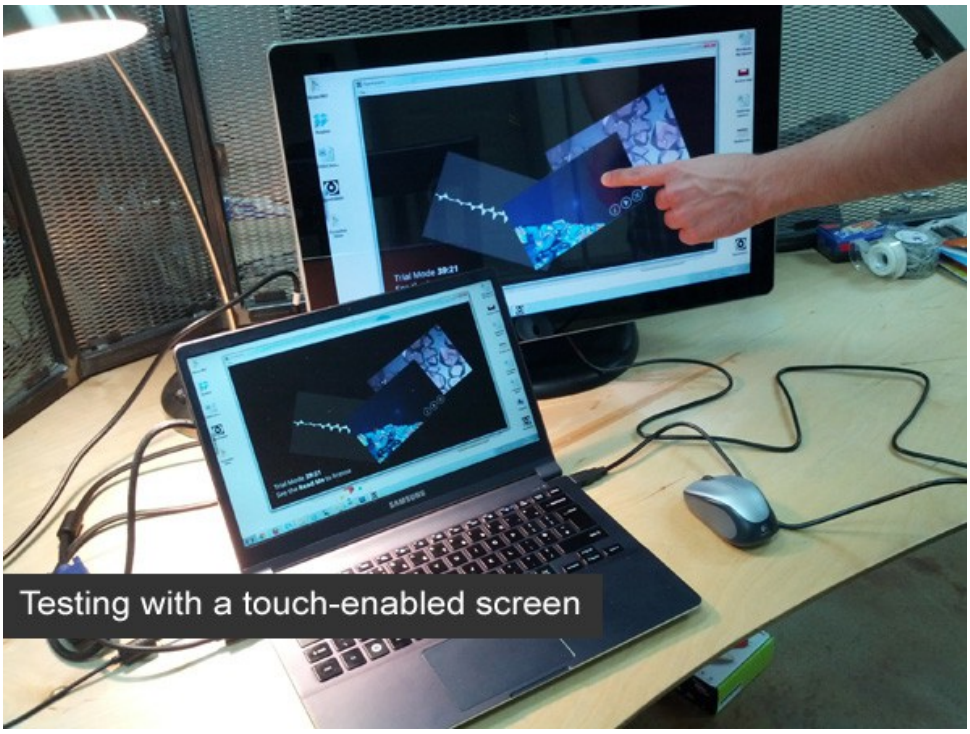
Step 3: Testing with a Touch-Enabled Device or Simulator

At this point, if you can see the media elements on the screen, then your installation of the Player was successful. To move the elements around on the screen and see the multitouch functionality of the Viewer, you have two options:

1) Connect your computer to a touch-enabled device

2) Turn on the Player's built-in simulator

If you already have a touch-enabled screen or table on which you will deploy your exhibit, then connect it to your computer (if you haven't already). As you can see in the picture below, you use your computer to open and edit CML files to build your exhibit and then open and test them on the touch device.

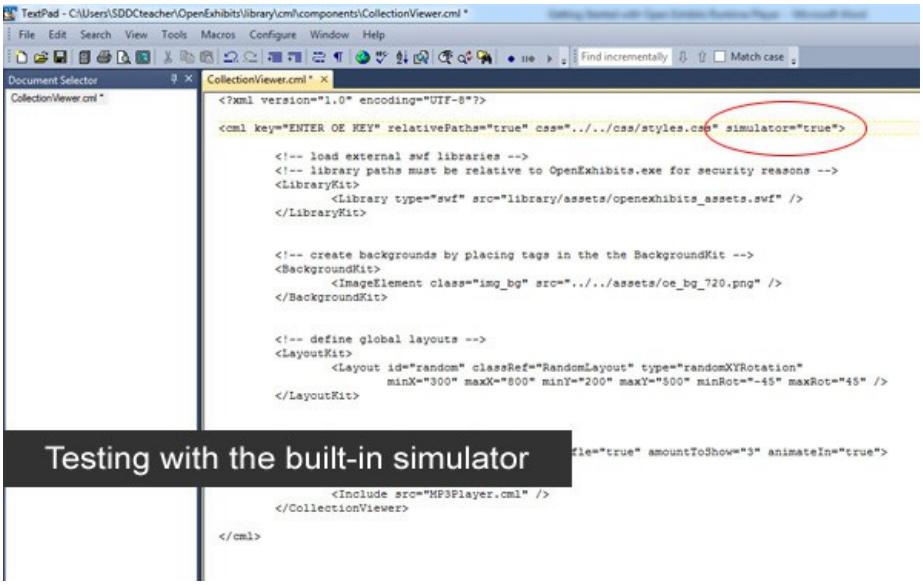


Activating the Simulator

If you don't have a touch-enabled device, the Player includes a Simulator which allows you to test on any standard computer. It uses mouse clicks to simulate touch events. A touch event is an action you do on a touch-enabled surface like tap, drag, rotate, or pinch and zoom. With the Simulator you can get an idea of how an exhibit will eventually work on a touch-enabled device.

Activating the Simulator is easy. Open a text editor and navigate to your Open Exhibits installation directory. Go to the "library > cml > components" subfolder. Open the CollectionViewer.cml in a text editor. If you are not a programmer, the file may look a little complicated, but don't worry. It is written in

Creative Markup Language (CML). You will only be making a simple edit now to enable the Simulator. We will come back to editing CML in the next part of the tutorial.



To activate the simulator, find the “cml” tag at the very top of the file. Inside the angle brackets you will see that some attributes like “key,” “relativePaths,” and “css” have already been defined. Add the following line inside the brackets to enable the simulator:

simulator=”true”

Once you have added it, the cml tag should look like the one below. You have just added something called an “attribute” in CML. There are many different attributes that control different features.

<cml key="ENTER OE KEY" relativePaths="true" css="../../css/styles.css" simulator="true">

Now save CollectionViewer.cml. Close the Open Exhibits Player and reopen it again. This time, you should be able to drag media elements by clicking them. Remember, that your clicks are only simulating users touching the screen. In order to deploy your application, you will need a touch-enabled device.

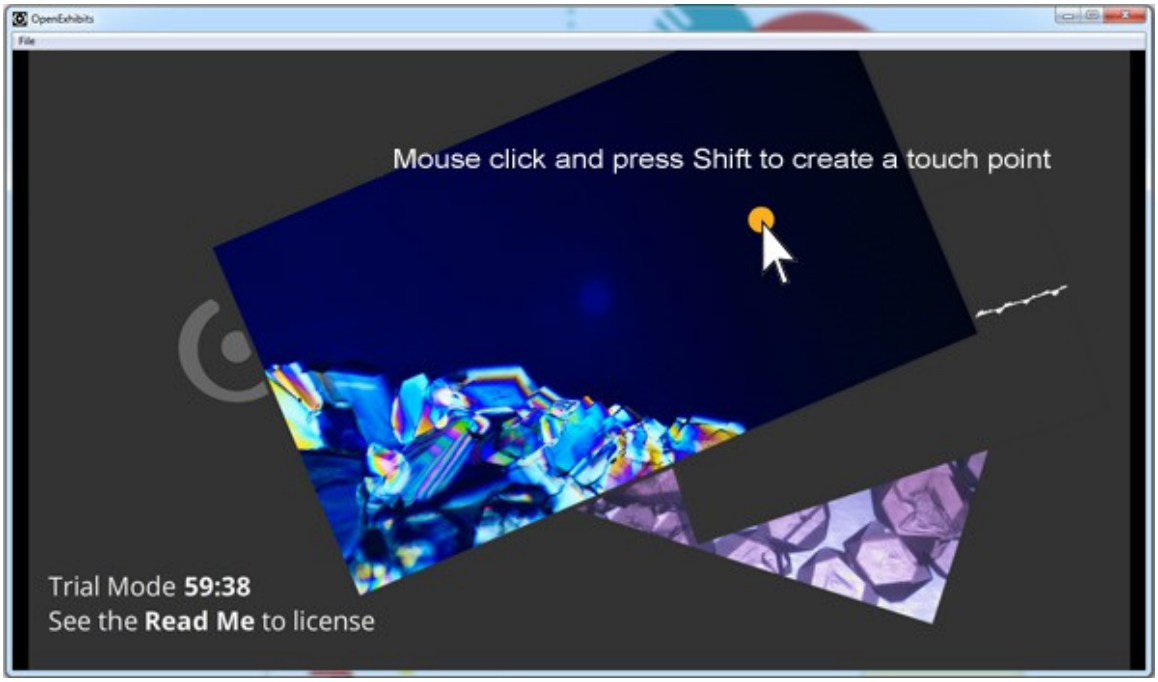
Working with the Simulator

The Simulator approximates the way a multi-touch device works. This means you can use it to drag, rotate and scale elements on the screen with your mouse. On a touch-enabled device, these actions would be done with hands and fingers. Moving elements in the Simulator is easy, just click and drag them.

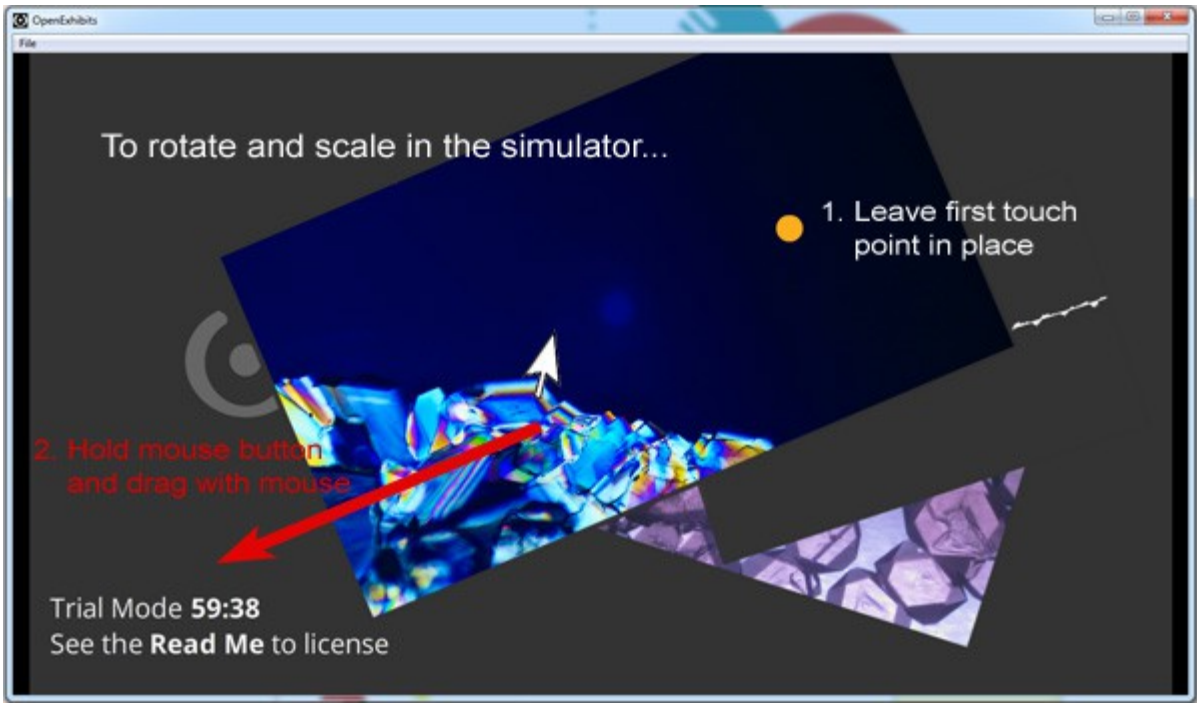
Rotating and scaling elements

Rotating and scaling is a bit more complicated, since on an actual touch device these actions require at least two fingers. To simulate multiple fingers touching the screen you set “touch points” in the

Simulator by holding the Shift key while clicking with your mouse. These actions create a touch point indicated by a gold-colored dot.

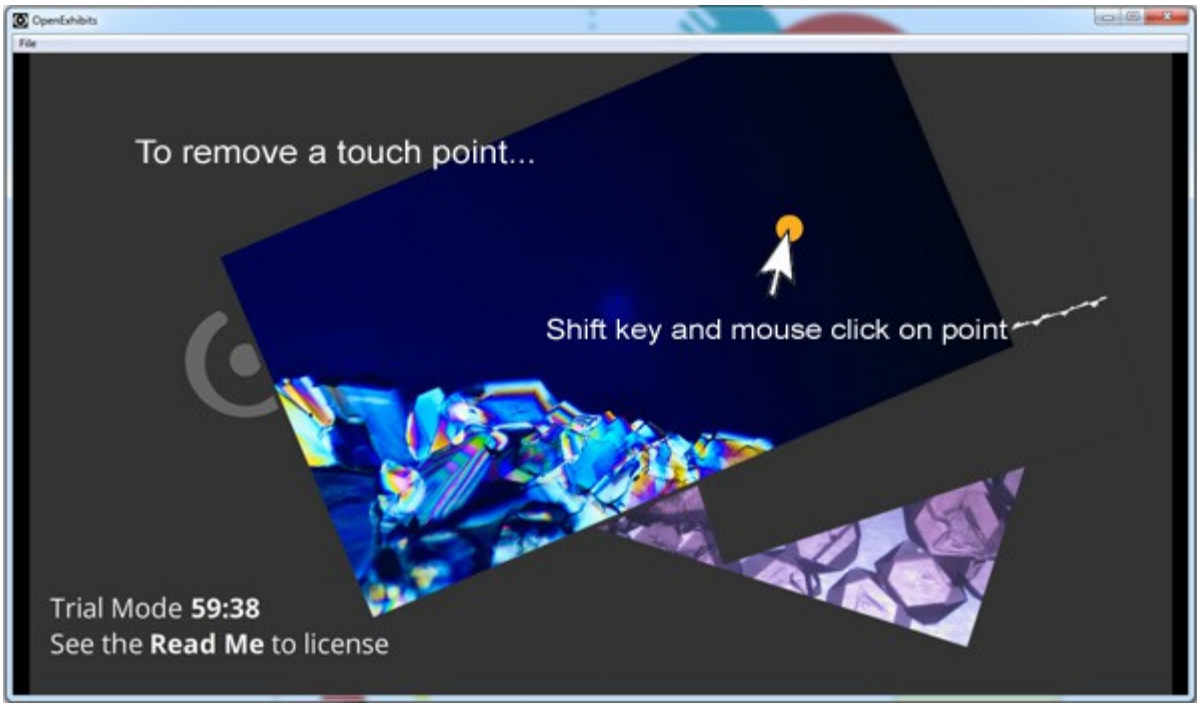


Once you have created a single touch point, you can let up on the Shift key and move your mouse pointer. The original touch point stays in place. You need at least two touch points to rotate or scale an element. Create a second point by clicking with your mouse. This time, instead of letting up, hold the mouse button down and drag with your mouse. The first touch point you created now becomes a pivot point for rotating and scaling the element.



Removing touch points

To remove a touch point, press Shift and click the gold dot with your mouse. The touch point has been successfully removed when the gold dot disappears.

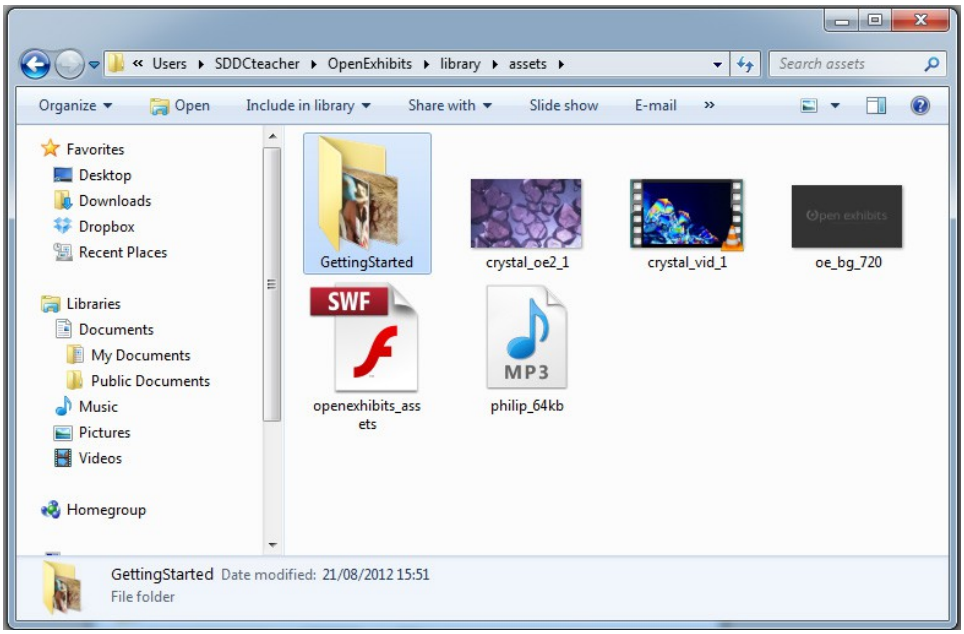

OpenExhibits
File
To remove a touch point...
Shift key and mouse click on point
Trial Mode 59:38
See the Read Me to license

Step 4: Customizing an Example Exhibit

The easiest way to get started creating your own exhibits is to customize an example. Now that you have seen the Collection Viewer, we will use that as the basis for your first exhibit. Authoring and editing of exhibits is done through editing Creative Markup Language (CML).

CML is based on XML, the Extensible Markup Language. All XML-based files have the same structure. Information and instructions are defined in “tags.” Tags have two parts: an opening and a closing marker. They are defined with angular brackets <OpeningTag> and forward slashes </ClosingTag> CML tags are nested inside each other. Think of them like a set of Russian dolls: The most general tag is at the top. Inside it are tags that define specific elements like images, sounds and videos that appear on the screen.

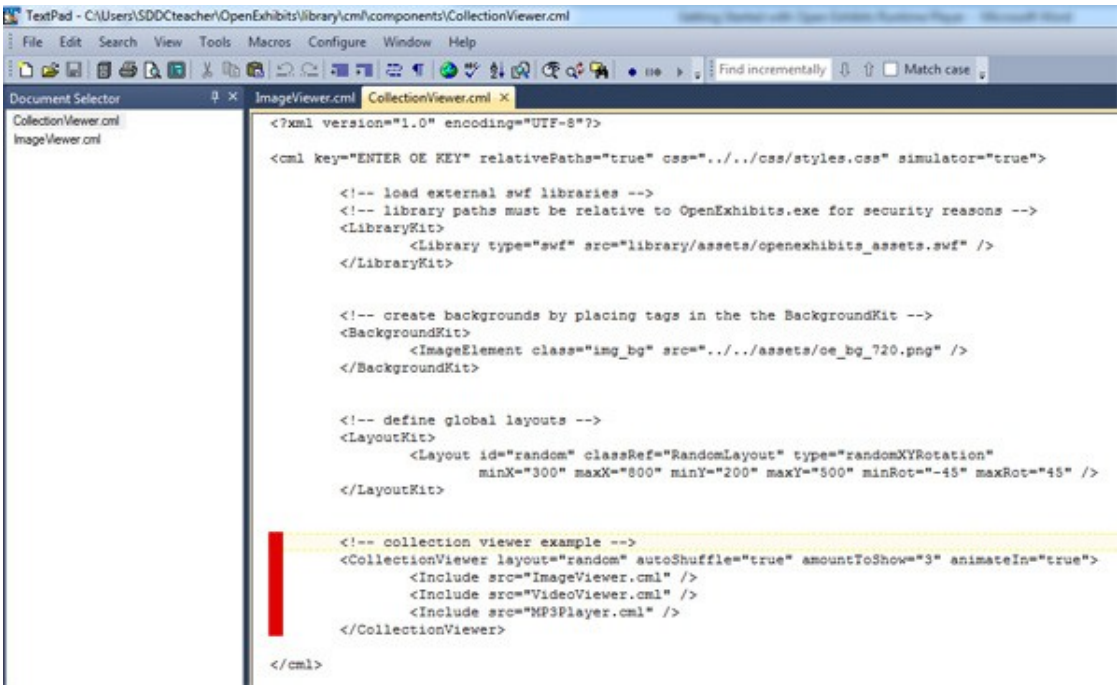
For this part of the tutorial, you will be working with a set of media files that were bundled with the Open Exhibits Player. These files are located in the library/assets subfolder of your Open Exhibits install directory (probably “Computer/Users/YourUserName/OpenExhibits/library/assets”).



Updating Media

Our first task is to swap out the existing media in the Collection Viewer, so you can learn how to incorporate your own media into an exhibit.

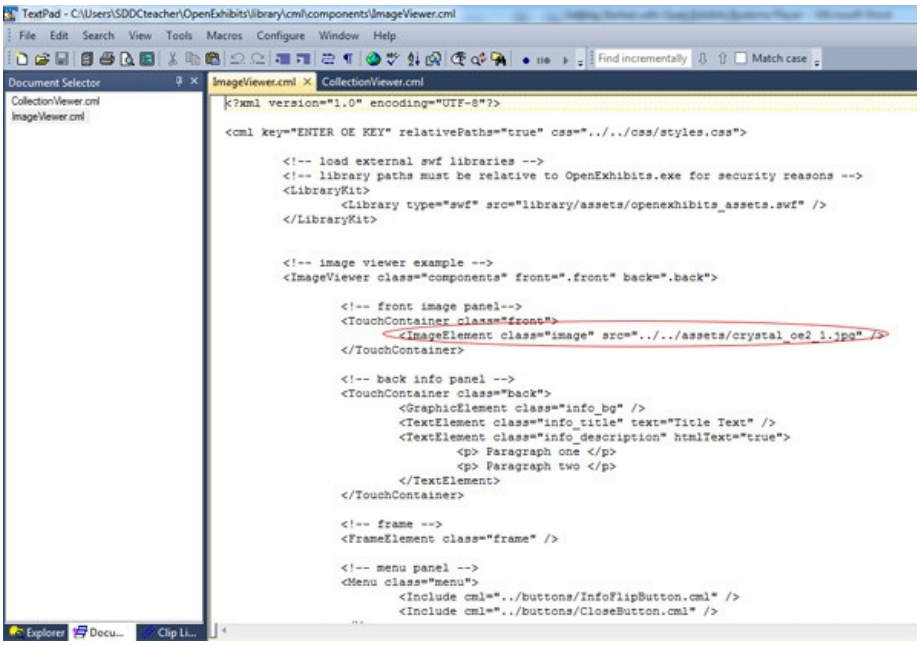
Media is updated by editing CML files. Open your favorite text editor. Click File > Open and navigate to the Open Exhibits installation directory (probably “Computer/Users/YourUserName/OpenExhibits”). Go to the library/cml/components folder and open CollectionViewer.cml.



At the bottom of the file is a section labeled “collection viewer example”. You see three lines that correspond to the three types of media displayed in the Collection Viewer: Image, Video and Audio (MP3). Media is defined in the CML files for those individual components. To display more media elements, we will need to add to this bottom section, but first let’s concentrate on replacing the existing three media elements.

Replacing an image

We will start with the still image. To do this, we will need to open the ImageViewer.CML file that is referenced in CollectionViewer.CML. Open ImageViewer.CML in your text editor. It is located in the same place as the CollectionViewer.CML, probably “Computer/Users/YourUserName/OpenExhibits library/cml/components.”

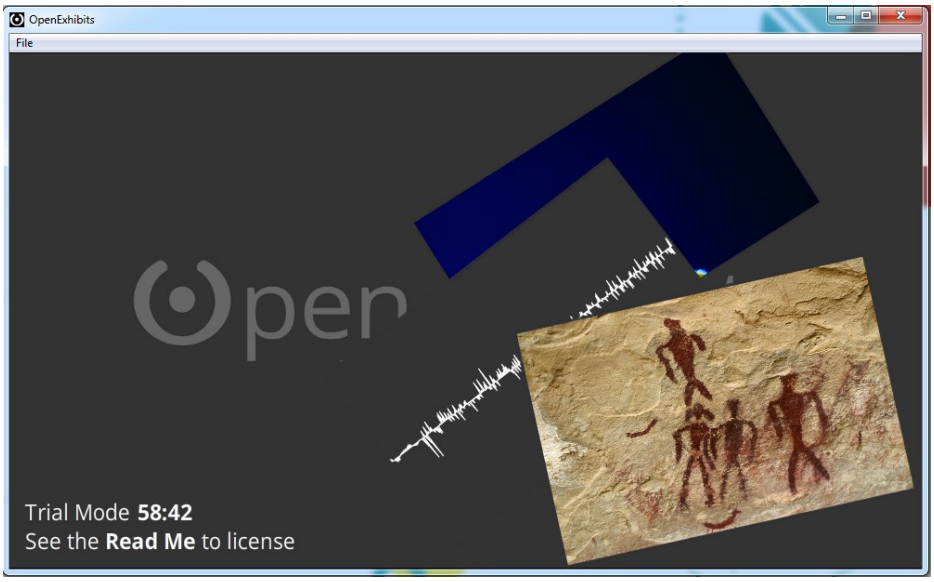


Look for TouchContainer tag. Inside it is an ImageElement tag which has a “src” attribute that specifies the image which is displayed in the Collection Viewer. At the moment, this is an image of crystals. Let’s replace this image with a one from our GettingStarted folder.

Change the “src” attribute of the ImageElement tag to show the path to “chaco-img-01.jpg” in the GettingStarted folder. The updated line should read:

<ImageElement class="image" src="../../assets/GettingStarted/chaco-img-01.jpg" />

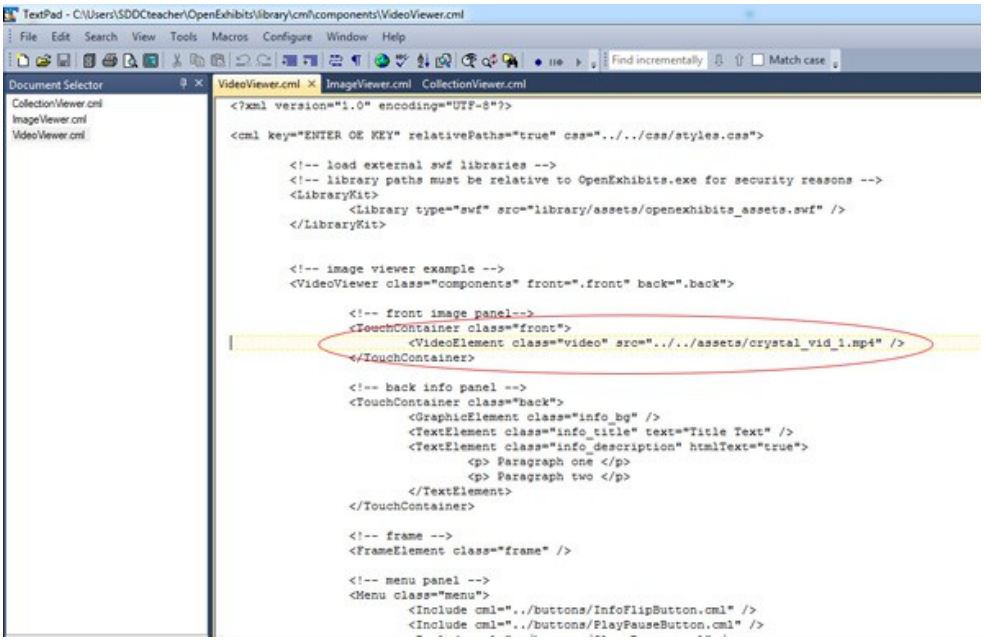
Let’s test to make sure the image has been replaced. If it is already running, close and restart the Open Exhibits Player. Open CollectionViewer.cml. An image of figures from Atlatl Cave in Chaco Canyon should have replaced the image of crystals.



Replacing video and audio

Use the same technique to replace the video and audio clips currently displayed in the Collection Viewer with assets from the GettingStarted folder. To replace the video, open VideoViewer.CML in your text editor and update the VideoElement tag:

<VideoElement class="video" src="../../assets/GettingStarted/chaco-vid-01.flv" />



Open Exhibits supports the following video formats:

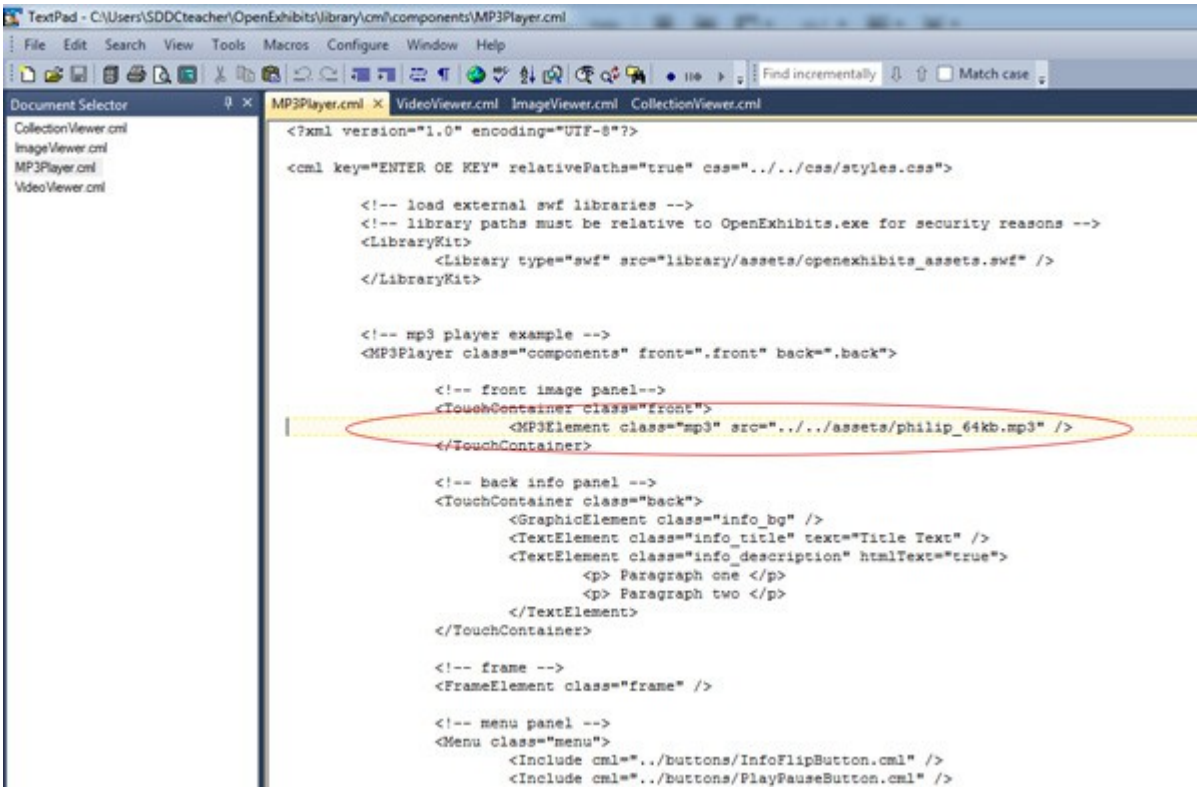
- FLV, F4V (Flash video)

- MP4
- MOV
- AVI

To replace the audio clip, open MP3Player.CML in your text editor and update the MP3Element tag:

<MP3Element class="mp3" src="../../assets/GettingStarted/ chaco-equinox-clip.mp3" />

To check your changes are correct, if it is already running, close and restart the Open Exhibits Player. As you did previously, open CollectionViewer.cml.



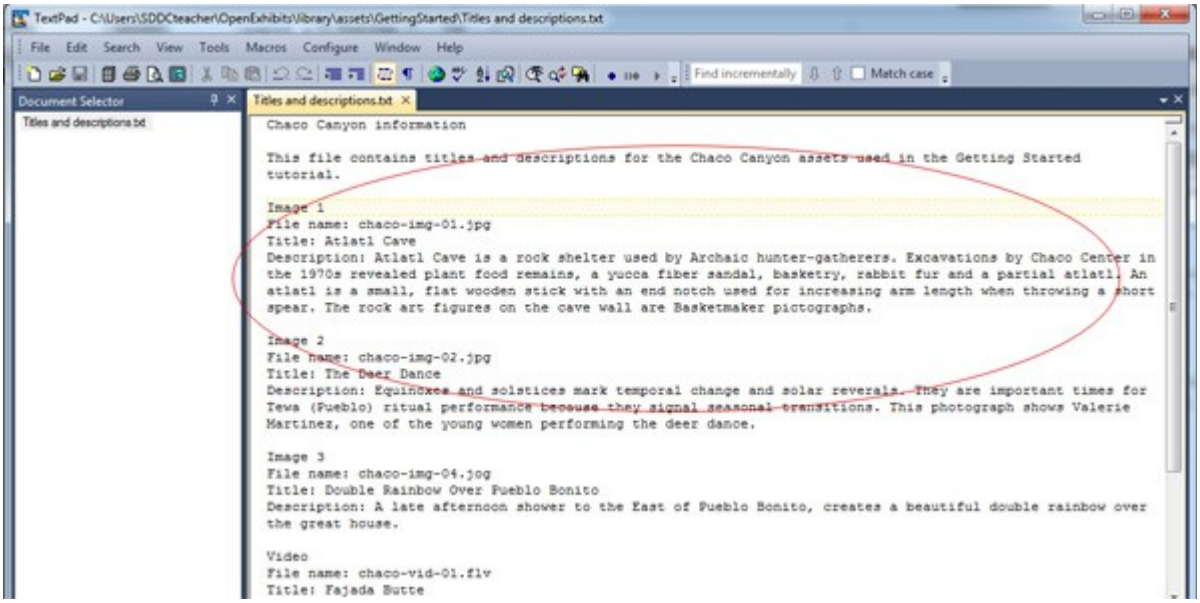
The MP3Player supports only files in MP3 format, however, you can also use WAV format files by using the WAVPlayer component.

Once you have replaced the original media with the new media assets from Getting Started, close and reopen the CollectionViewer.CML in the Open Exhibits Player. The result will look something like the screenshot below. If media are missing, double check the paths and file names in the CML files you edited. Make sure you have included the “Getting Started” folder in the path after “assets.”

Adding Text and Metadata

Now that you have successfully updated your media, let's change the information (or metadata) about the media that displays when the "i" button is pressed. You may have noticed that when you tap a media element on the screen, a menu appears. Pressing the information button "i" brings up a text layer with information about the image.

We are going to replace the placeholder text with the title and description of this image. The text for all the media for this tutorial can be found in the text file Titles and Descriptions in the Getting Started folder (probably “Computer/Users/YourUserName/OpenExhibits/library/assets/Getting Started/”). Open the Titles and Descriptions file in your text editor. Look for the section for Image 1, Atlatl Cave.



Now, open ImageViewer.CML and locate the “back info panel.” Think of each media element as a card with a front and back. The visual element is on the front (image, video, etc.) and the information or metadata is on the back.

```
<!-- back info panel -->
        <TouchContainer class="back">
                <GraphicElement class="info_bg" />
```

```
            <TextElement class="info_title" text="Title Text" />
            <TextElement class="info_description" htmlText="true">
                <p> Paragraph one </p>
                <p> Paragraph two </p>
            </TextElement>
        </TouchContainer>
```

You can see that inside the back panel for the image are two text elements. The first is a title, the second is a description. Go back to the Titles and Descriptions text file in your editor. Copy the title for Image 1 into ImageViewer.CML. Replace “Title Text” with the actual title you have copied “Atlatl Cave.”

The description works a little differently. Instead of pasting the content inside the TextElment tag like you did for the title, put the copied text inside the opening and closing markers. You are replacing the text inside the two paragraph <p> tags. When you are finished the block of CML above should look like this:

```
    <!-- back info panel -->
        <TouchContainer class="back">
            <GraphicElement class="info_bg" />
            <TextElement class="info_title" text=" Atlatl Cave" />
            <TextElement class="info_description" htmlText="true">

                <p>Atlatl Cave is a rock shelter used by Archaic hunter-gatherers. Excavations by Chaco Center in the 1970s revealed plant food remains, a yucca fiber sandal, basketry, rabbit fur and a partial atlatl.</p>

                <p>An atlatl is a small, flat wooden stick with an end notch used for increasing arm length when throwing a short spear. The rock art figures on the cave wall are Basketmaker pictographs.</p>

            </TextElement>
        </TouchContainer>
```

Save ImageViewer.CML. Close and reopen CollectionViewer.CML in the Player. Now when you press on the “i” button on the menu, the “back” of the image shows the correct title and description for the image.

Now that you have seen how updating text works, try replacing the text for the video and audio elements on the screen. Remember you will need to edit the correct CML files where those elements are defined. For the video, edit VideoViewer.CML, for the audio, edit MP3Player.CML. Once you have added all the media, the result should look something like the image below.

Step 4: Customizing and Styling the Example Exhibit

The next step in customizing your exhibit is to change the look and feel to reflect the theme, content or house style of your exhibit. In Open Exhibits, changing the design of an exhibit is done by editing CML and Cascading Style Sheets (CSS). If you have designed a website before, you will probably be familiar with CSS. If it is entirely new to you, you may want to complete a basic CSS tutorial before proceeding.

Changing and Styling the Background Image

We will start with replacing the background image to complement our Chaco Canyon content. Open the file CollectionViewer.CML in your text editor. Look for the BackgroundKit tag. Inside it is an ImageElement like we saw in the Image Viewer.

```
<BackgroundKit>
<ImageElement class="img_bg" src="../../assets/oe_bg_720.png" />
</BackgroundKit>
```

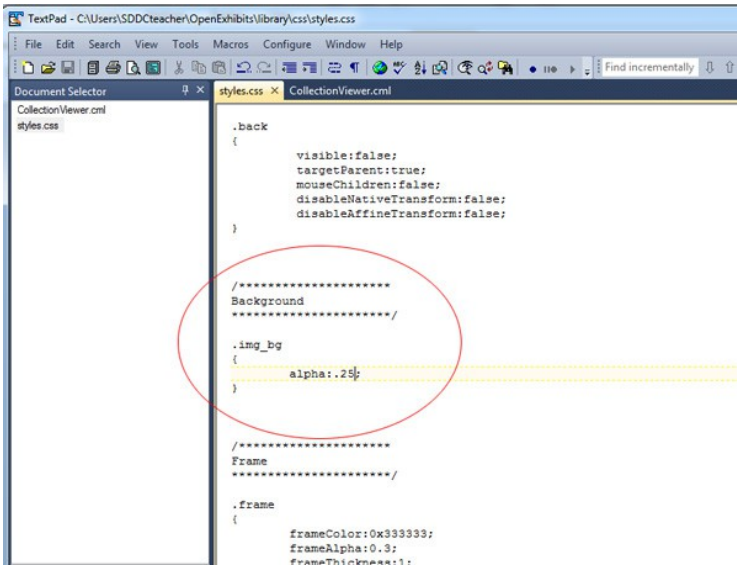
Update the background by changing the “src” value to one of the background images in the Getting Started folder.

```
<BackgroundKit>
<ImageElement class="img_bg" src="../../assets/Getting Started/background-01.jpg" />
</BackgroundKit>
```

Once you have updated the background, close and restart the Open Exhibits Player. The new background should appear behind your media elements.

You may notice that the background looks dim, as if there is a layer on top of the image. This effect has been applied in the Style Sheet linked to the CML document. Let's modify this effect to see how you can style images and other elements with CSS.

Open the styles.css file in the library/css folder of your Open Exhibits install directory (probably "Computer/Users/YourUserName/OpenExhibits"). Scroll down to the "Background" section. In CSS, the alpha attribute sets the opacity of elements. Here, it determines how bright our background image appears on the screen.



The current value is .25 which is 25% opaque. To make the image display at full opacity, change the alpha value to 1.0 which is 100% opaque.

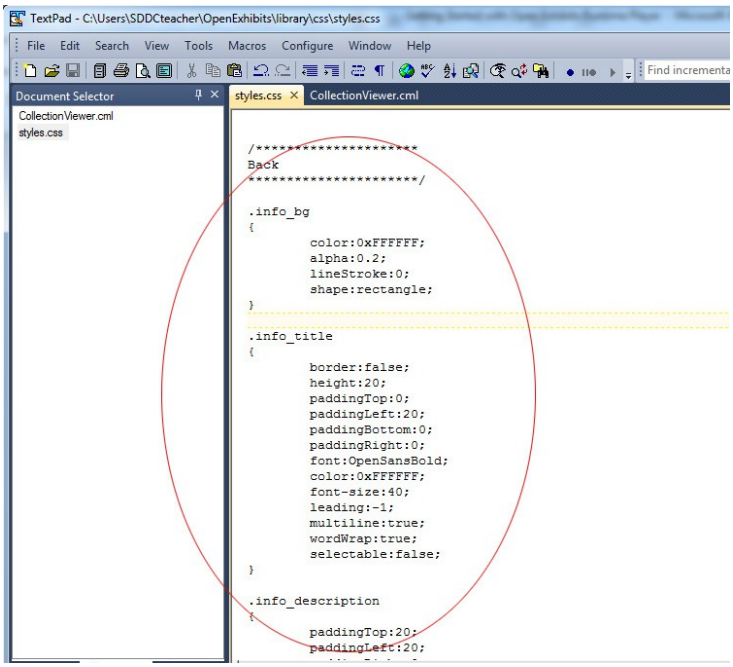
```
/**********************
Background
**********************/
.img_bg
{
alpha:1.0;
}
```

Once you have changed the alpha value, save the styles.css file. Close and reopen the Collection Viewer in the Player to show the changes you just made. The background should now appear brighter than before.

Style the Text

To accommodate more text or to change the look of your exhibit, you may want to change the colour and font of the text. We already added titles and descriptions to the media elements, but now let's change the style of the text on the Atlatl Cave image. The text could be smaller and a darker color would make it more legible.

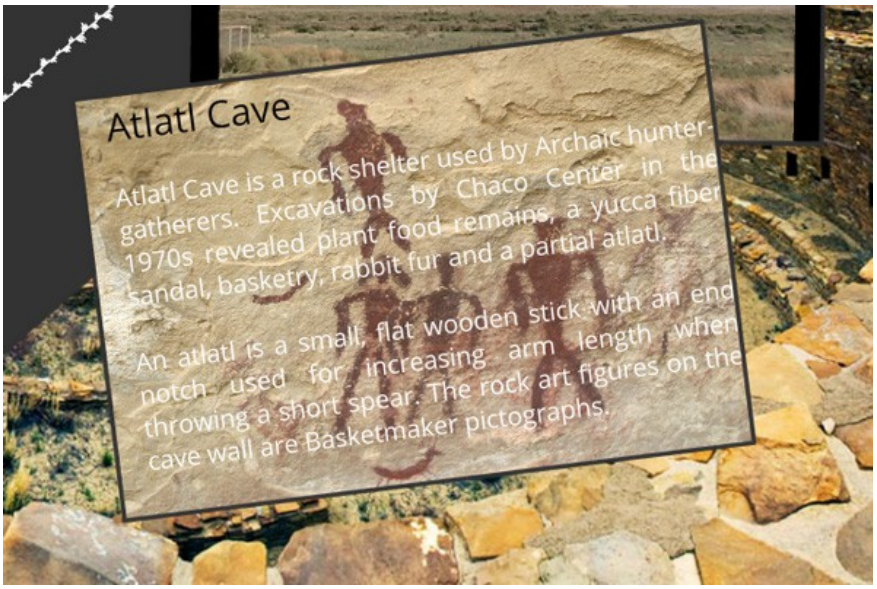
Open the styles.css file in the library/css folder of your Open Exhibits install directory (probably "Computer/Users/YourUserName/OpenExhibits"). Look for the "Back" section. Remember that each media element has a front and back. The front is the visual content (image, video, etc.) and the back contains the title and description.



To change the size and color of the title we will change .info_title. The two properties we need to change are color and font-size. Let's change the font from bold to regular, the color to black and reduce the size. CSS uses hexadecimal values for colors. You can easily find a color chart to convert RGB values to hexadecimal values. Update the values under .info_title with those below.

font:OpenSansRegular;
color:0x00000;
font-size:30;

Save the styles.css file. Close and reopen the Collection Viewer in the Player to show the changes you just made. Click on the "i" to display the image title. The title is black in color and smaller than before.



Do the same for the description, which can be found under .info_description in the styles.css file.

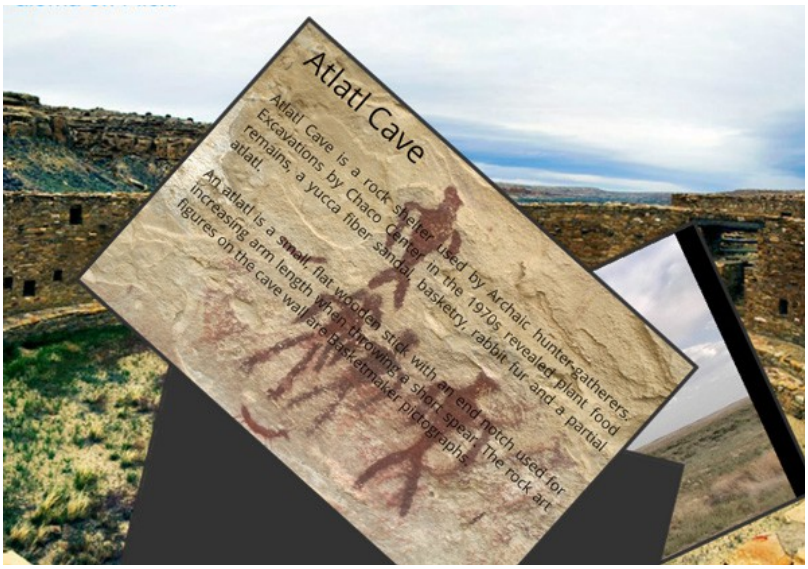
```
        wordwrap:true;
        selectable:false;
}

.info_description
{
        paddingTop:20;
        paddingLeft:20;
        paddingRight:0;
        font:OpenSansRegular;
        color:0xFFFFFF;
        font-size:20;
        leading:-1;
        multiline:true;
        wordWrap:true;
        textAlign: justify;
        border:false;
}
```

Update the values for color and font-size to make the text fit neatly into the frame. Try the values below first and then experiment by changing the values on your own.

color:0x00000;
font-size:15;

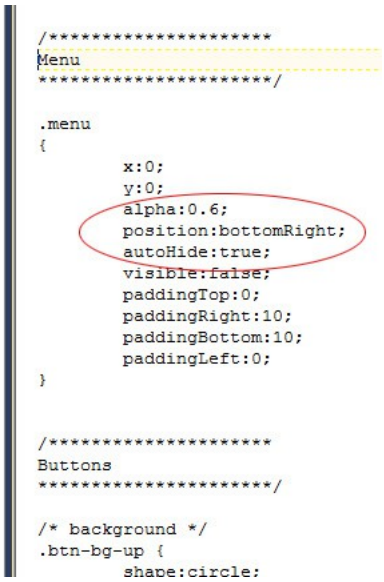
Each time you make changes, save the styles.css file. Close and reopen the Collection Viewer in the Open Exhibits Player and click on the “i” to display the text panel.



Restyling and Changing Menu Buttons

As you saw when we changed the titles and descriptions, there are buttons which appear in the bottom right corner of each media element. The buttons show what you can do with the media element and so there are different buttons for different media. The Video Player below has information, play and close buttons.

Let's change the position of these buttons and restyle them. Both of these changes are made by editing CSS. Open the styles.css file in the library/css folder of your Open Exhibits install directory (probably "Computer/Users/YourUserName/OpenExhibits"). Look for the "Menu" section. The location of all the buttons is set together in the Menu element.



```
/*********************
Menu
**********************/

.menu
{
        x:0;
        y:0;
        alpha:0.6;
        position:bottomRight;
        autoHide:true;
        visible:false;
        paddingTop:0;
        paddingRight:10;
        paddingBottom:10;
        paddingLeft:0;
}


/*********************
Buttons
**********************/

/* background */
.btn-bg-up {
        shape:circle;
```

To move the menu to the top right corner of the element to the top left corner, change the value of the "position" property as shown below.

```
.menu
{
        x:0;
        y:0;
        alpha:0.6;
        position:topLeft;
        autoHide:true;
        visible:false;
        paddingTop:0;
        paddingRight:10;
        paddingBottom:10;
        paddingLeft:0;
}
```

Save the styles.css file. Close and reopen the Collection Viewer in the Open Exhibits Player to show the changes you just made. Click on the video element to display the button menu, which should now appear in the top left corner.

Let’s try restyling the buttons by changing their transparency. As you can see in the screenshot above, the buttons are now partially transparent. This is because the “alpha” property in styles.css is set to 0.6, which means the entire menu is 60% opaque. (Remember that we saw the alpha property when we styled the background image.)

To change the transparency of the buttons, return to the “Menu” section of styles.css. Update the alpha property to 1.0 as shown below.

```
.menu
{
	x:0;
	y:0;
	alpha:1.0;
	position:topLeft;
	autoHide:true;
	visible:false;
	paddingTop:0;
	paddingRight:10;
	paddingBottom:10;
	paddingLeft:0;
}
```

Save the styles.css file. Close and reopen the Collection Viewer in the Player to show the changes you just made. Click on any of the media elements to display the restyled buttons, which should now be completely opaque.

The style of buttons is set on three levels. We have just made some global changes to all the buttons by changing the Menu. To change the color, size and shape of the buttons however, we need to edit the Buttons section of styles.css. Return to your text editor and find the Buttons section.

```
/**********************
Buttons
**********************/

/* background */
.btn-bg-up {
        shape:circle;
        radius:20;
        lineStroke:1.5;
        color:0x282E33;
}

.btn-bg-down {
        shape:circle;
        radius:20;
        lineStroke:1.5;
        color:0x495E67;
}
```

At the top of this section we can set values for all the buttons together. Let's change the background color from grey to blue to complement the look and feel of our exhibit. CSS defines colors according to hexadecimal values. There are many websites that will convert RGB colors to their hex equivalents. Update the color property as shown below to the hex value for light blue (3399FF).

```
/* background */
.btn-bg-up {
        shape:circle;
        radius:20;
        lineStroke:1.5;
```

```
        color:0x3399FF;
}
```

Save the styles.css file. Close and reopen the Collection Viewer in the Player to show the changes you just made. Click on any of the media elements to display the restyled buttons, which should now be light blue.

We have just seen how to restyle all the buttons at once using by changing Menu and Buttons sections of the CSS. Now let's change the font color of a single button. Return to styles.css and scroll down to find the info button and flip button.

```
/* info button */
#info-fg-up {
        alpha:1;
        x:15;
        y:8;
}

#info-fg-down {
        alpha:1;
        x:15;
        y:8;
}

/* flip button */
#flip-fg-up {
        alpha:1;
        x:7;
        y:7;
}

#flip-fg-down {
        alpha:1;
        x:7;
        y:7;
}
```

These properties define the info button on two different sides of the “card” on which the media, title and description is shown.

When the media element first appears on the screen, the “i” button is displayed. If the user clicks the button to display the title and description, then the button changes to an arrow. To change the color of the info button “i” to dark blue, add a color property to info-fg-up values as shown below.

```
/* info button */
#info-fg-up
{
        alpha:1;
        x:15;
        y:8;
        color:0x000099;
}
```

Save the styles.css file. Close and reopen the Collection Viewer in the Player to show the changes you just made. Click on any of the media elements to display the restyled info button.

Not all buttons function this way. Some are embedded in the Player’s Flash library and must be changed in CML. This is the case with the flip arrow that toggles to the other side of the “card” where the image is displayed without text or description. To change the color of the flip arrow, we will edit a button CML file.

Open the FlipButton.CML file in the library/cml/buttons folder of your Open Exhibits install directory (probably “Computer/Users/YourUserName/OpenExhibits”). Look for the Container tag where the flip-up button is defined.

<Container id="flip-up">
<GraphicElement id="flip-bg-up" class="btn-bg-up" />
<SWFElement id="flip-fg-up" class="btn-fg-up" classRef="org.openexhibits.assets.Flip" />
</Container>

These lines reference a library of vector graphics that are loaded when the CollectionViewer launches. To change the foreground color of the flip arrow to dark blue, add a color attribute to the second tag.

<SWFElement id="flip-fg-up" class="btn-fg-up" classRef="org.openexhibits.assets.Flip" color="0x000099" />

Save the FlipButton.CML file. Close and reopen the Collection Viewer in the Open Exhibits Player to show the changes you just made. Click on any of the media elements to display the restyled arrow button.

There is one more way to restyle buttons. If you have your own images, you can swap out the existing menu buttons. Do this by removing the reference to the Flash library and replace it with your own ImageElement. Let's replace the info button with our own image.

Open the InfoButton.CML file in the library/cml/buttons folder of your Open Exhibits install directory (probably "Computer/Users/YourUserName/OpenExhibits"). Look for the Container tag for info-up.

<Container id="info-up">

~~<GraphicElement id="info-bg-up" class="btn-bg-up" />~~

~~<SWFElement id="info-fg-up" class="btn-fg-up" classRef="org.openexhibits.assets.Info" />~~

</Container>

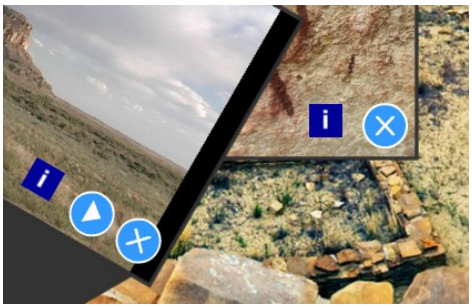
Remove the two lines which reference the Flash library and replace them with the line below.

<Container id="info-up">

<ImageElement id="info-bg-up" class="btn-bg-up" src="../../assets/GettingStarted/button-info.jpg" />

</Container>

Save the CML file. Close and reopen the Collection Viewer in the Open Exhibits Player to show the changes you just made. Click on any of the media elements to display the new button.



Styling frames around media

To improve the look of the media elements, you can add, remove or style the frames around them. You may have noticed that the elements in our example have black borders around them. This is a frame referenced in the CML files for individual components. Its style is set in CSS. Let's style the frame around the Atlatl Cave image.

Open the styles.css file in the library/components/css folder of your Open Exhibits install directory (probably "Computer/Users/YourUserName/OpenExhibits"). Look for the Container tag for the Frame element.

```
/**********************
Frame
**********************/

.frame
{
        frameColor:0x333333;
        frameAlpha:0.3;
        frameThickness:1;
        frameShape:rectangle;
        scaleX:0.995;
        scaleY:0.995;
}
```

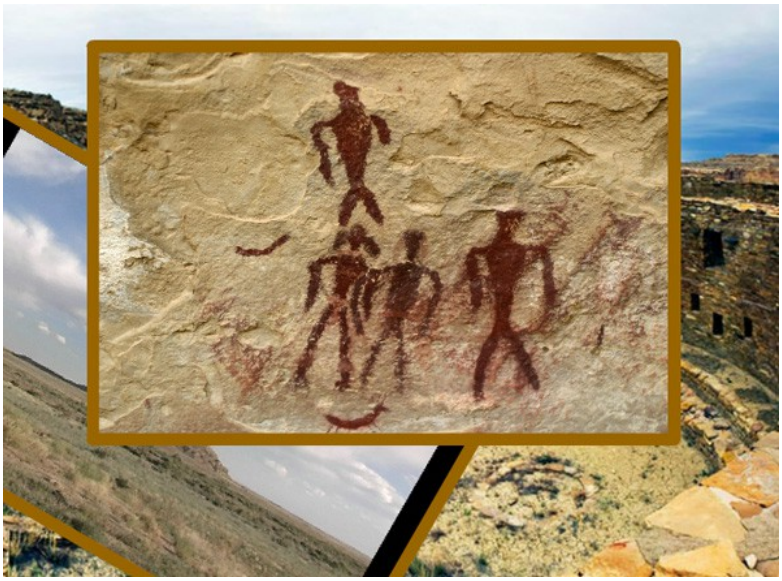
Change the frame styles with the values below.

```
/**********************
Frame
**********************/

.frame
{
        frameColor:0x 996600;
        frameAlpha:1.0;
        frameThickness:5;
        frameShape:rectangle;
        scaleX:0.995;
```

```
        scaleY:0.995;
}
```

Save styles.css. Close and reopen the Collection Viewer in the Player to show the changes you just made.



In some cases you may want to remove the frame entirely. If you have assets with transparent backgrounds for example, they may look strange with frames around them. Let's remove the frame around the Atlatl Cave image.

Open the ImageViewer.CML file in the library/components/cml folder of your Open Exhibits install directory (probably "Computer/Users/YourUserName/OpenExhibits"). Find the frame element.

```
<!-- frame -->
<FrameElement class="frame" />
```

Add the visible attribute to the FrameElement tag as shown below

```
<FrameElement class="frame" visible="false"/>
```

Save ImageViewer.CML. Close and reopen the Collection Viewer in the Player to show the changes you just made.

Congratulations!

You have successfully completed the Getting Started tutorial and are well on your way to building your first multitouch exhibit. Move on to the next tutorial for advice on how to design your exhibit, or if you are a developer, choose one of the more advanced tutorials on setting up your development environment to work with ActionScript. If you have problems or questions about working with Open Exhibits, please post to the support forums. Once you have completed an exhibit, please consider sharing your experience with the Open Exhibits community through a blog post or exhibit download.