



Getting Started with Xerte

An Introduction to Interactive Development

Julian Tenney
June 2007

Introduction.....	3
Installing Xerte.....	3
Starting a New Project	5
The Xerte Interface	5
Viewing your Work	7
Getting Help.....	8
How a Learning Object Flows	9
A Note on Scripting	10
Importing Media	10
Visual Development.....	11
For Flash Developers	12
The Xerte Icons.....	13
Learning Object Icon	13
Interface Icon	13
Page Icon.....	14
Text Icon	15
Graphic Icon.....	16
Movie Icon	17
Sound Icon	18
Drawing Icon	20
Script Icons	21
Interaction Icons.....	22
Interaction Responses	24
How Interaction Responses Work	24
Buttons	24
Radio Buttons.....	25
Checkboxes	26
Combo Boxes.....	27
Hot Objects	28
Hot Spots.....	30
Text Entry	31
Target Areas.....	32
Time Limit	33
Tries Limit	34
Events.....	34
Framework Icons	34
Framework Entry Frame	35
Decision Icons.....	36
Accessibility.....	36
Creating Models and Templates	36
Models.....	36
Deploying your Projects	37
Publishing a Project	37
Packaging a Project.....	38

Introduction

Xerte is a suite of tools for the rapid development of interactive learning content. In developing Xerte we have sought to make it very easy to perform simple, common tasks, yet possible to do anything you want. Xerte seeks to provide a focus on the types of problems and situations that developers of interactive learning content frequently encounter.

Xerte provides a visual, icon-based authoring environment that allows learning objects to be easily created with the minimum of scripting. Functionality that would be time-consuming to develop from scratch in other tools can be created very quickly in Xerte.

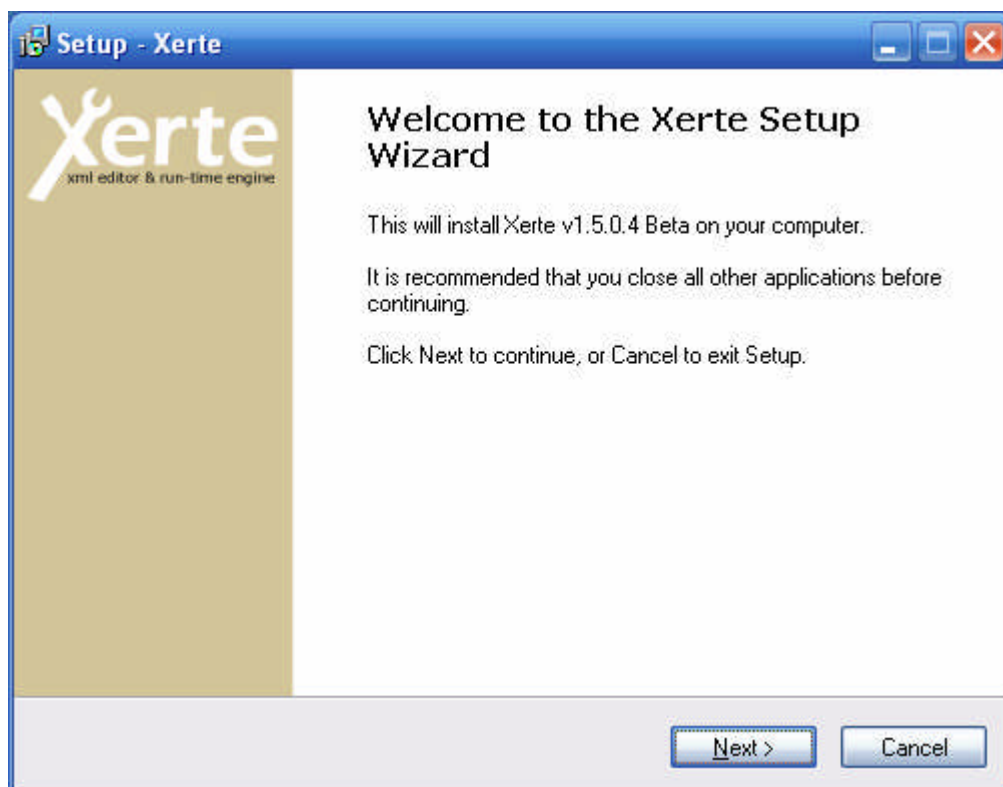
If you don't write code yet, you will be able to use Xerte to integrate text, graphics, animations, sounds and video, create simple interactivity, and deliver it in an accessible interface.

If you write some code, you will be able to create more complex structures and sophisticated interactivity. You'll be able to customise the default interface and develop your own interfaces and navigation systems for your learning objects.

If you write lots of code you'll be able to develop powerful components and even extend the Xerte authoring interface with your own tools.

Installing Xerte

For the latest build of Xerte, visit <http://www.nottingham.ac.uk/xerte>. Download the installer, and run it. Xerte will be installed onto your computer.



The 'read me' file presented at the end of the installation process will tell you about bug fixes and new features and functionality.

Xerte is updated frequently. You can subscribe to the RSS feed on the Xerte site to ensure you are informed when new releases have been made available – usually about once a month, though the actual files may be updated more frequently than that.

You can also subscribe to the Xerte mailing list on the web site. This list is for letting us know any issues you have found, reporting bugs, asking for new features and generally discussing development with other Xerte users. It's a great place to ask any questions you have, or to seek advice on development issues.

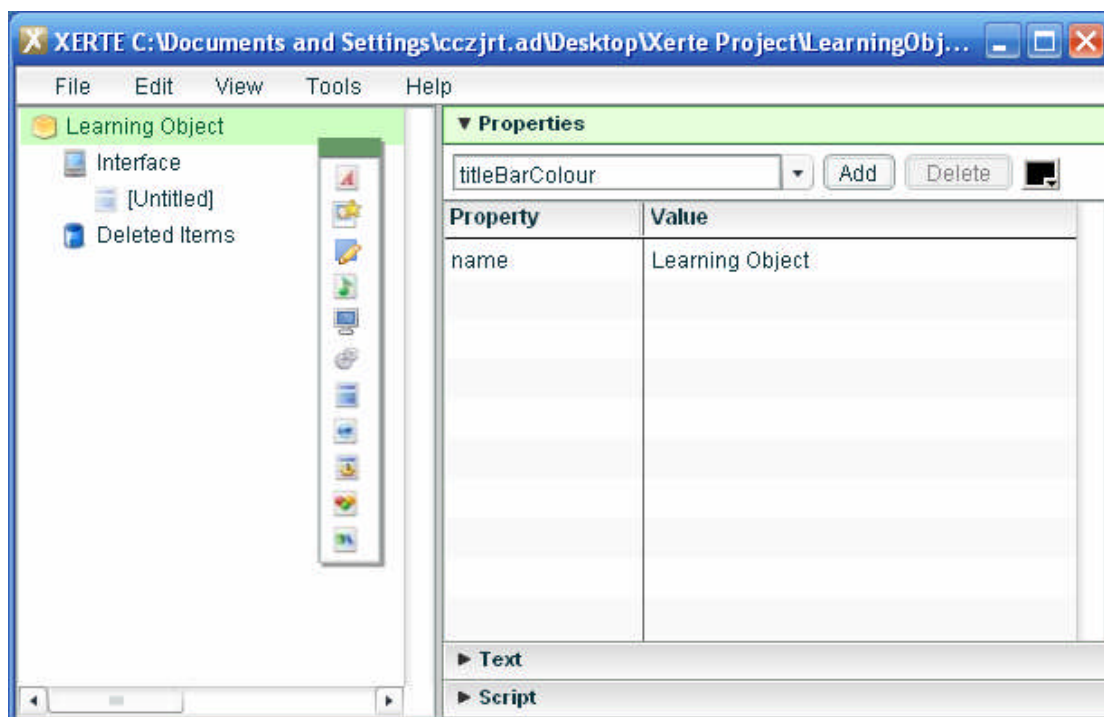
Starting a New Project

To start a new project, create a new folder for the project on your computer. It's a good idea to keep all projects in separate folders as the project will eventually comprise a number of files.

Launch Xerte and choose File – New. Save the file in the folder you have just created.

To create a new Xerte project easily, you can right-click in the folder you just created and choose New > Xerte File from the context menu.

The Xerte Interface



On the left-hand side of the interface is the document tree. You construct a learning object by adding icons to this tree from the icon palette. There are eleven different types of icons. Some are used for adding media to a learning object, whilst others provide structure and interactivity. The icons are shown in the icon palette. To add icons to the tree you can:

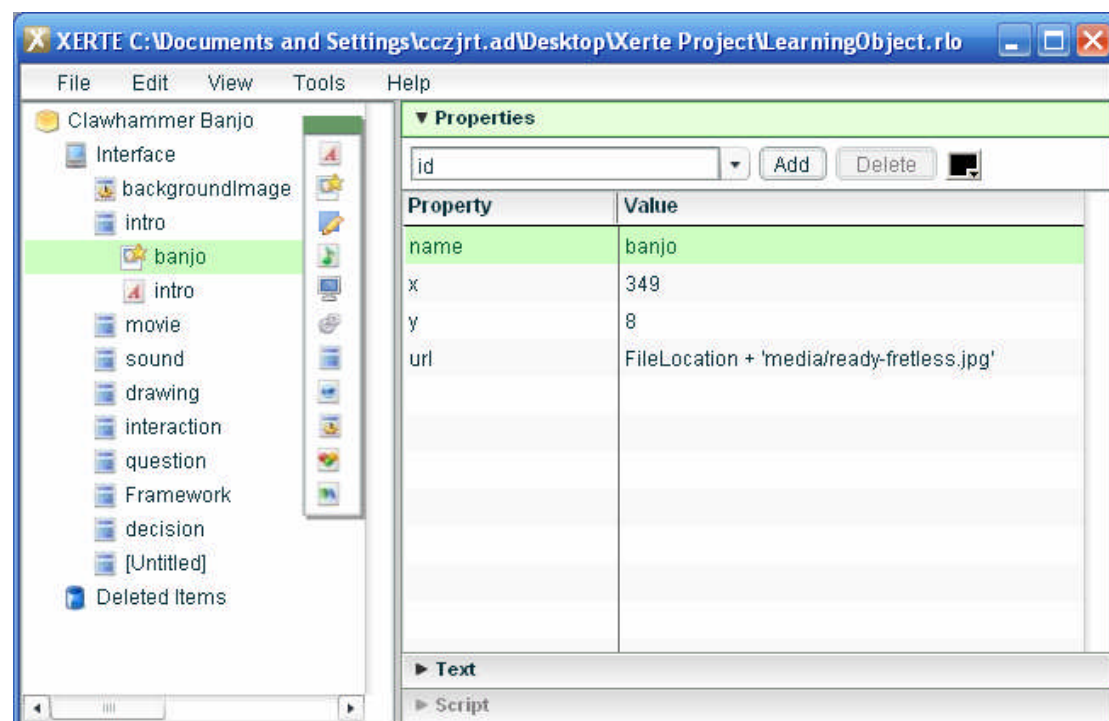
- Drag the type of icon you want to add from the icon palette to the place in the document tree where you want to put the icon.
- Select the icon in the document tree to which you want to add a new icon and double click the type of icon you want to add in the icon palette.
- Right click the icon in the tree you want to add a new icon to, and choose the type of icon from the context menu.

When you start a new project, the document tree will contain four icons: The Learning Object Icon, the Interface Icon, an empty Page icon and Deleted Items. As you develop the project you will add further page icons to the interface icon to structure your content and add icons to each page to integrate media and create interactivity.

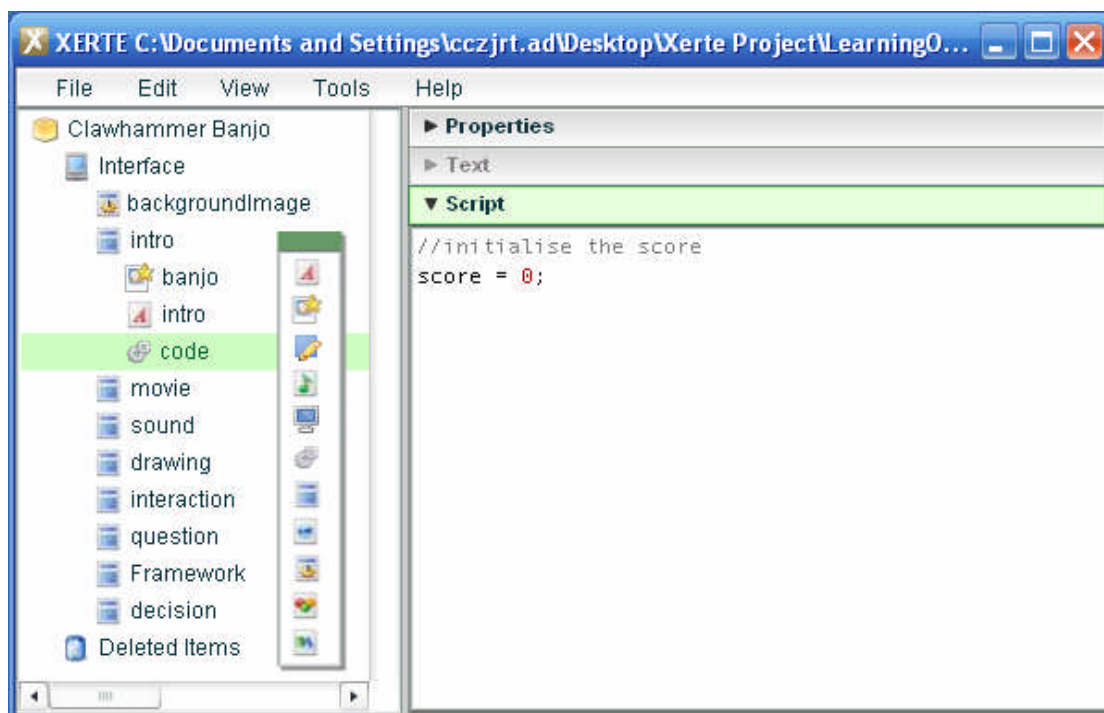
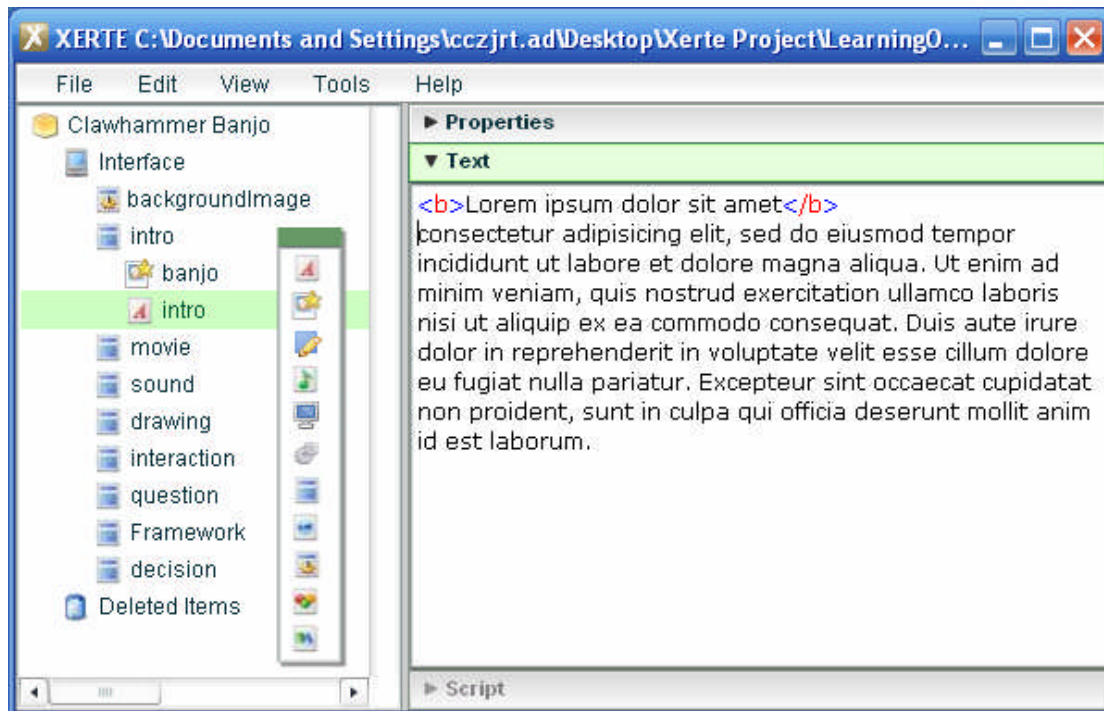
You can easily move icons around in the document tree by dragging and dropping them. Xerte will prevent you from placing icons where they do not belong. If you delete an icon it will be placed in the Deleted Items, from where you can retrieve it later if you need to.

As you add icons to the tree, you set their properties to configure the functionality your learning object will provide. Properties for icons are set in the property inspector on the right hand side of the interface. Selecting an icon in the document tree will show its properties in the property inspector. You use the property inspector to set the values of the icon's properties, and it makes it easy to refer to the current values of properties as you work.

Icons have some mandatory properties, which must always be set, and some optional properties which you may need. Optional properties are added from the drop-down list at the top of the properties inspector, which will show a list of the available optional properties.

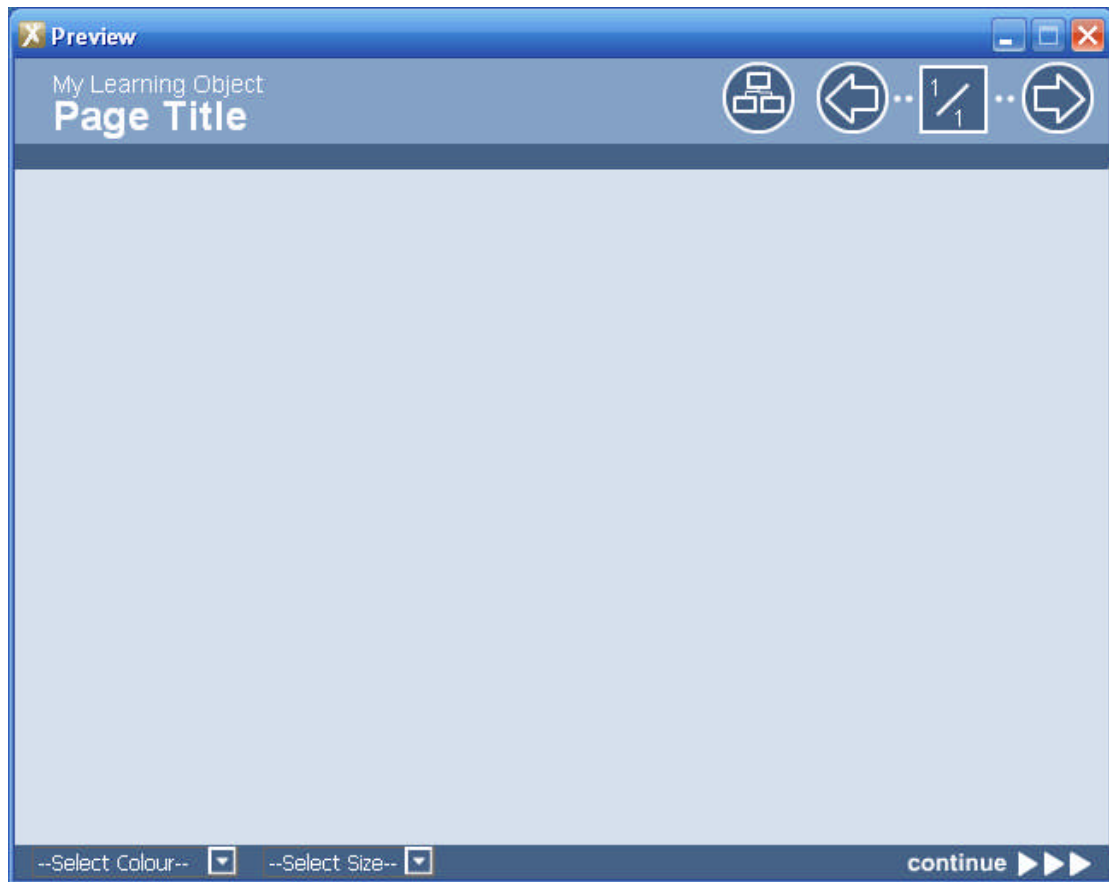


Below the properties inspector are two panels. You use the text panel to enter any text that Xerte will display in text icons, and you use the Script panel to enter any code that Xerte will execute to provide rich interactivity.



Viewing your Work

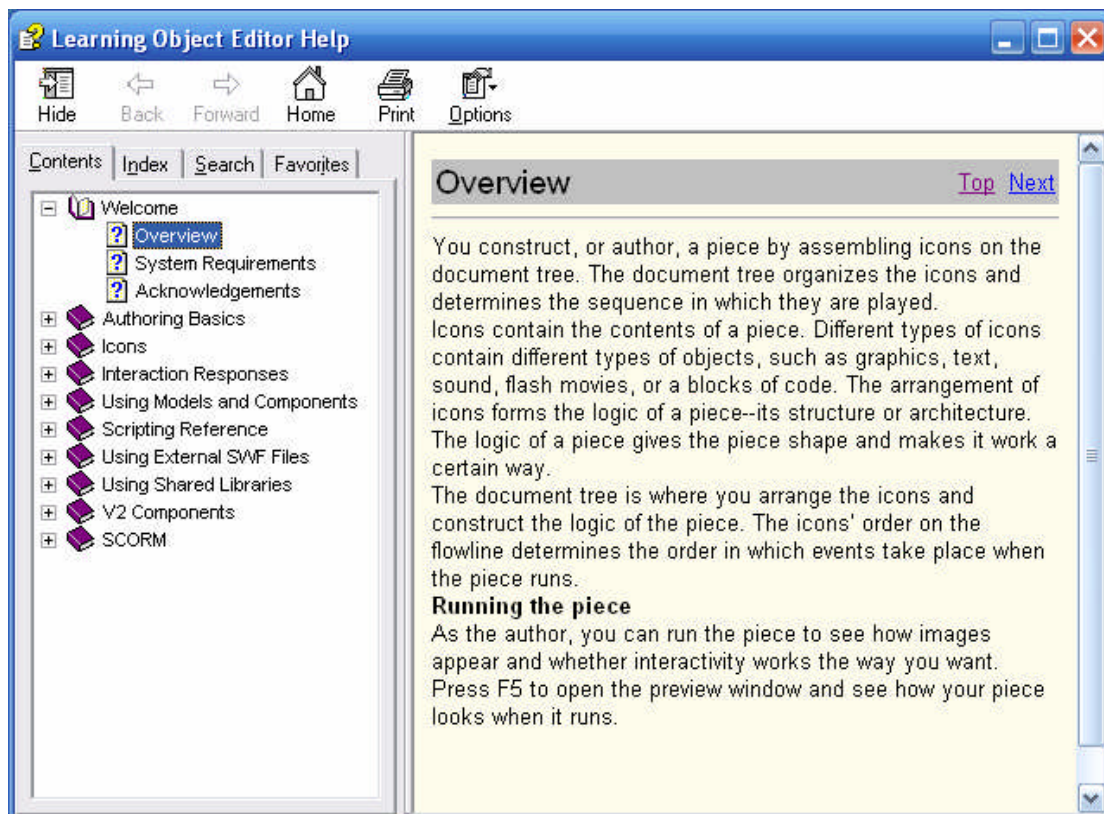
As you work, you will want to run the piece frequently to check it is functioning as you intend. To run the piece, press F5. Your project will run.



Xerte provides a default interface that will suit many situations, and provides accessible functionality. Developing learning objects that use the default interface will save a lot of work in creating the equivalent functionality. The default interface can be turned off if it doesn't suit the project you are trying to build, allowing you to develop a custom interface using Xerte or Adobe Flash.

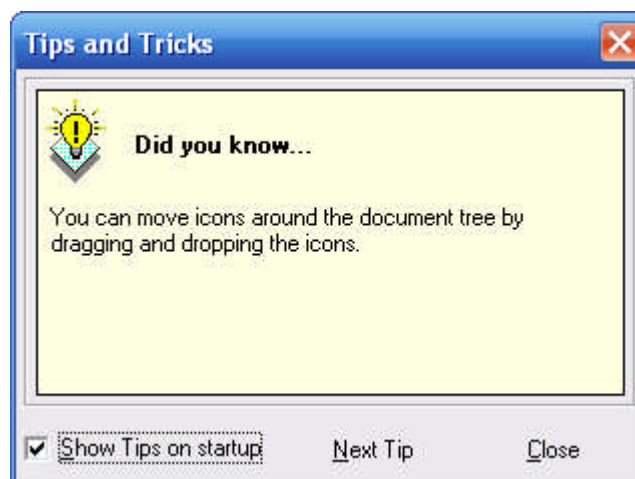
Getting Help

As you get to know Xerte, you will find the help file useful. The help file can be accessed from the help menu. The help file is a work in progress, but does provide a description of every icon and the icon's properties, methods and events, as well as other useful information.



You might want to refer to the help file as you read through this document. It is not intended to reproduce material from the help file here.

Reading through the tips presented when Xerte starts up is also recommended. You can view these tips at any time, using the Help menu.



How a Learning Object Flows

The document tree represents the structure of the learning object. When the learning object runs, Xerte's run-time engine begins at the top of the tree and starts flowing down the tree. As the flow moves down the tree, icons are created and added to the stage. The flow will then either continue to the next icon or halt, depending on the type of icon encountered and its properties.

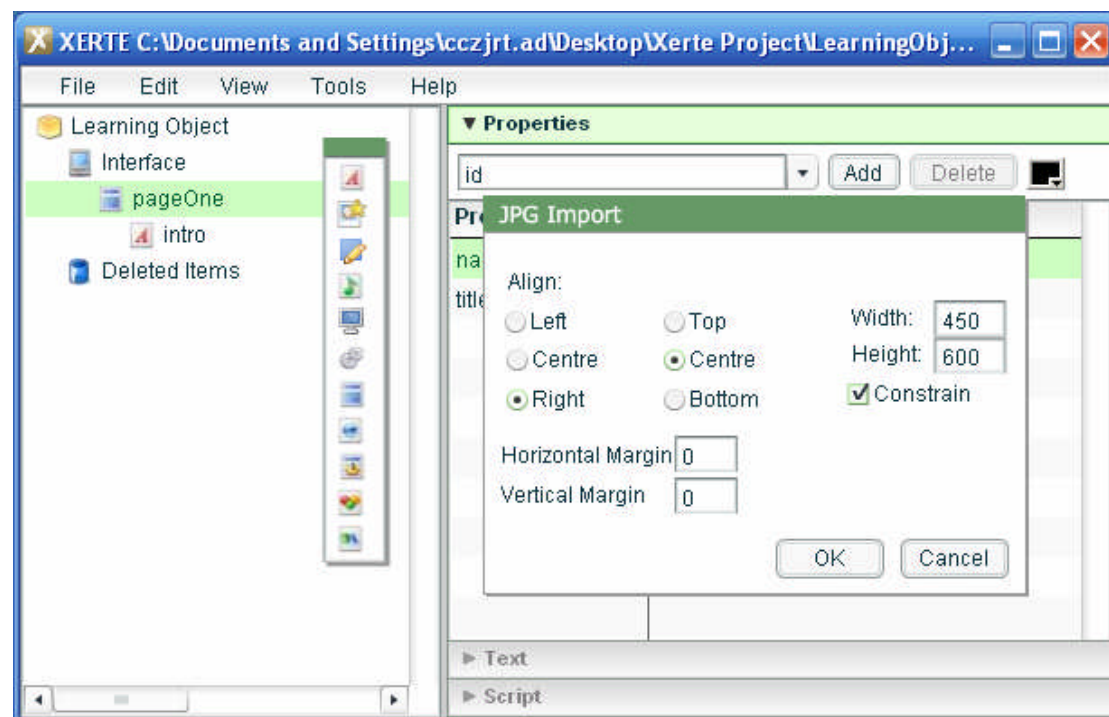
A Note on Scripting

There is an appetite for tools that allow non-programmers to produce powerful and sophisticated learning objects without writing any code. As far as I know, no tool has yet met that goal. I believe scripting is the only way to efficiently create complex, bespoke interactivity that does exactly what *you* want it to. Xerte has been designed with scripting users in mind. It has been my experience that tools that rely on a suite of pre-configured templates never quite do what I need. Perhaps I can't modify the way feedback is given after question, or the navigation systems don't suit the type of content I want to develop.

You can produce content with Xerte without doing any programming, but there are limitations as to how far you can go. By learning a little scripting, you become empowered to create exactly what you want. Scripting is not difficult, and the scripting language in Xerte is deliberately simple and easy to get to know as it is based on Flash's ActionScript and JavaScript. In fact, Xerte is a great place to start to learn some scripting as you will be able to apply your new skills to other languages that use the same syntax as Xerte.

Importing Media

You can quickly import graphics, flash animations, flash video and sounds into the your project by simply selecting the page into which you wish to import media in the document tree and dragging the file into Xerte. You will be presented with an import dialogue, allowing a number of choices depending on the type of media being imported.



You can also import media using the 'Import media' command on the file menu. When media is imported into a project, a copy of the file you import is placed into the project's media folder. If this folder doesn't exist, it will be created for you.

Visual Development

It is not always convenient to enter the co-ordinates of icons into the properties inspector. To visually position items on the stage, select the icon you want to position and press SHIFT-F5 to preview the learning object with the selected icon highlighted on the stage. You will be able to move and resize the icon using your mouse, and you'll be able to select other items on the stage and reposition / resize them too.

As you add icons to the tree and configure them you will want to view the resulting content frequently. To preview the learning object and jump straight to the icon you have selected in the document tree, press CTRL-F5.

To jump straight to an icon and adjust its size / position, hold down CTRL-SHIFT-F5.

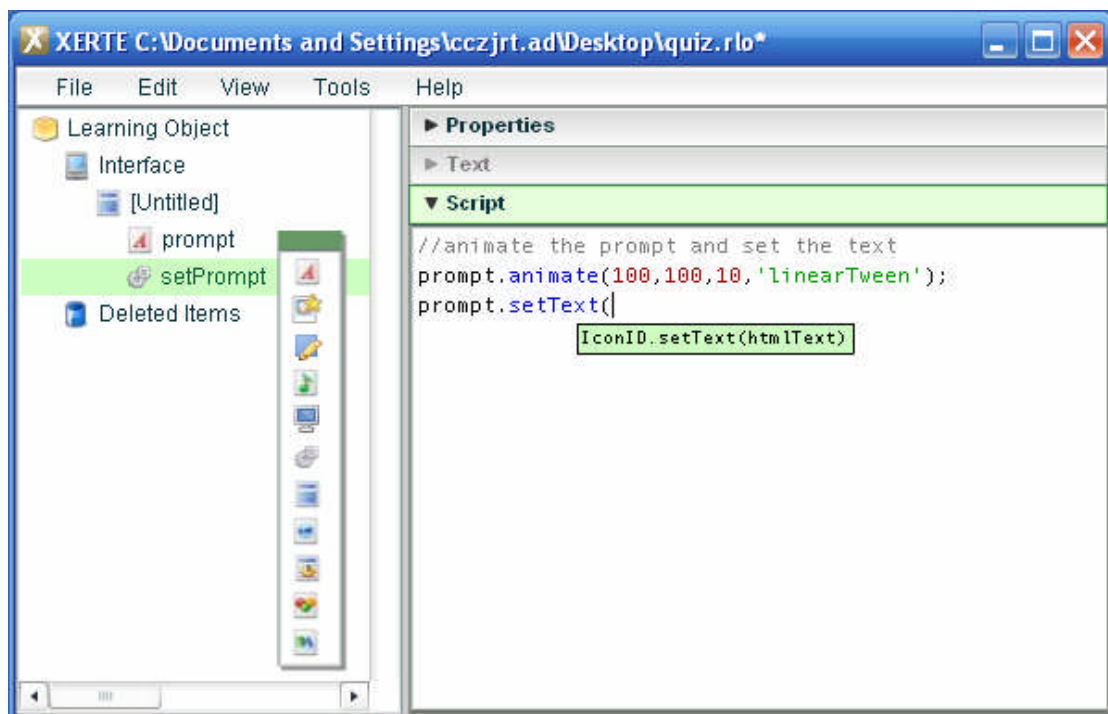


Here, the size and position of the picture can be set with the mouse. You can also move the image with the arrow keys on the keyboard for fine control. As the image is moved, its co-ordinates are shown to the top left.

For Flash Developers

Xerte really comes alive when you use it in conjunction with Flash. Xerte can take care of the authoring tasks, whilst Flash can be used to create animations, interactivity, custom buttons and interface elements, video and all the other things that Flash is good at. Flash-based content is easily integrated with Xerte.

If you are a Flash developer, it will be useful to know from the outset that every icon in the document tree becomes a MovieClip when the Learning Object runs. This is important because it means that icons can do all the things that MovieClips can do in Flash, as well as some extra things that all Xerte icons can do, as well as the specific things that only this type of icon can do. If you have an understanding of object-oriented programming, it will make sense to you that all icons inherit from a base class, ICON, which in turn inherits from MovieClip. When you give an icon an *id* property, you can address that icon with script, and call its methods and set its properties. For a comprehensive list of all the properties, methods and events for the icons, see the help file.



If you are a Flash developer, it will also be useful for you to understand that an rlo file is really an xml file. When the learning object runs, a flash movie loads the XML file and parses it, creating the learning object dynamically from the information found in the XML file. Each node in the XML file corresponds to a movie clip in the Flash player.

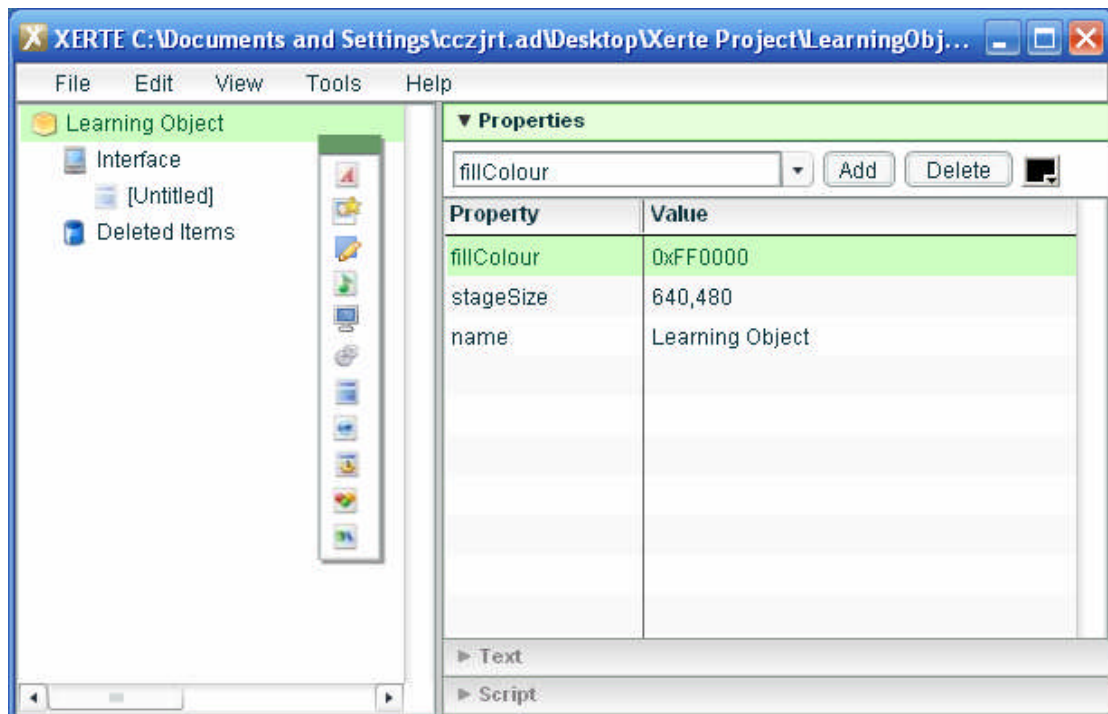
The Xerte Icons

The icons that you add to the document tree allow complex structures of content and sophisticated interactivity to be developed very quickly. As you add icons to the piece it is a good idea to give them a name, so you can find icons in the tree easily later. This becomes very important as your piece becomes more complex.

Learning Object Icon

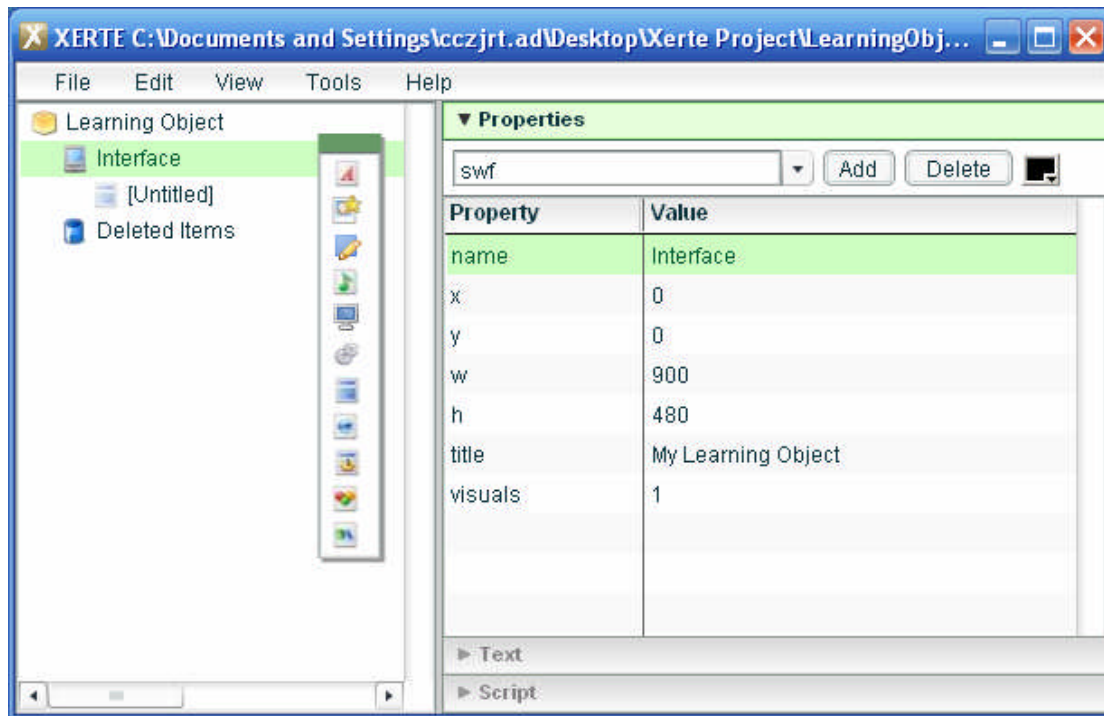
The Learning Object is the first icon in the document tree. It contains all the other icons in the piece, and there is only ever one Learning Object icon in a piece. If you are creating a simple piece, you probably don't need to worry about the Learning Object icon at all.

The Learning Object icon is where you set all the global style properties for the learning object to determine its look and feel, such as the size of the stage, or the default colour of text.



Interface Icon

You will use the Interface icon to determine whether the Learning Object will use Xerte's default interface, or your own custom interface.

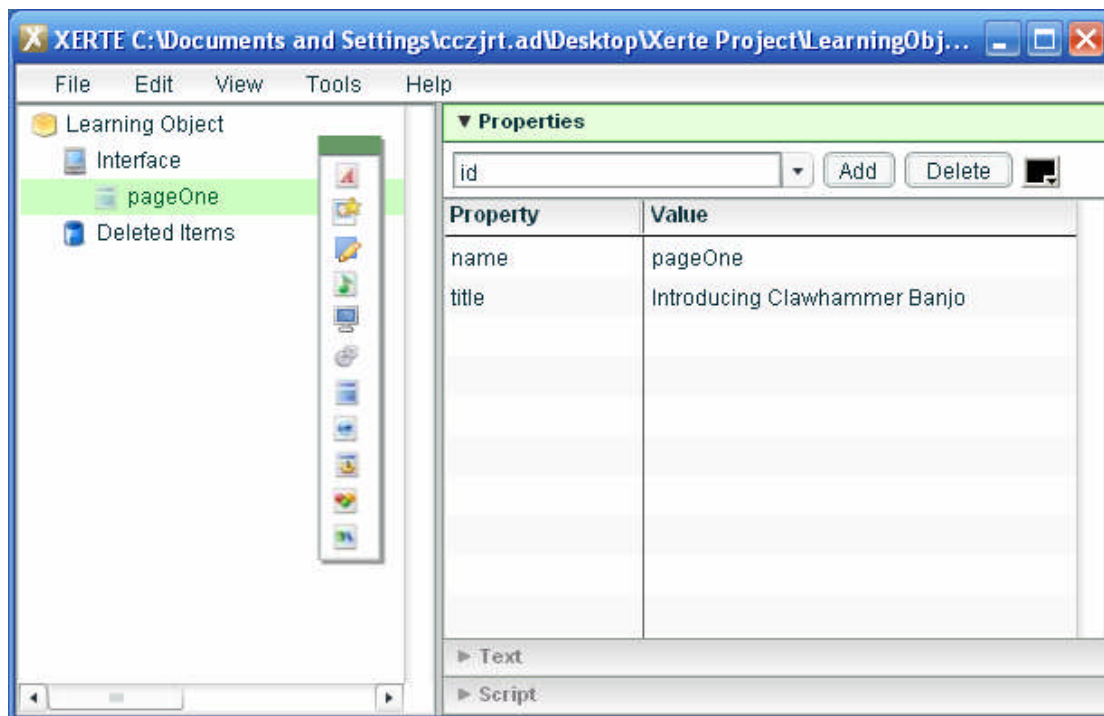


You can choose whether to use the default interface or not. If you turn off the default interface, you can develop a custom interface either using Xerte or Flash. If you specify a Flash movie to serve as the interface it should be based on the framework template in Xerte's templates/frameworks folder.

The optional properties allow you to link to a help file for the project. If you do so, the default interface will contain a help button that will launch the help file.

Page Icon

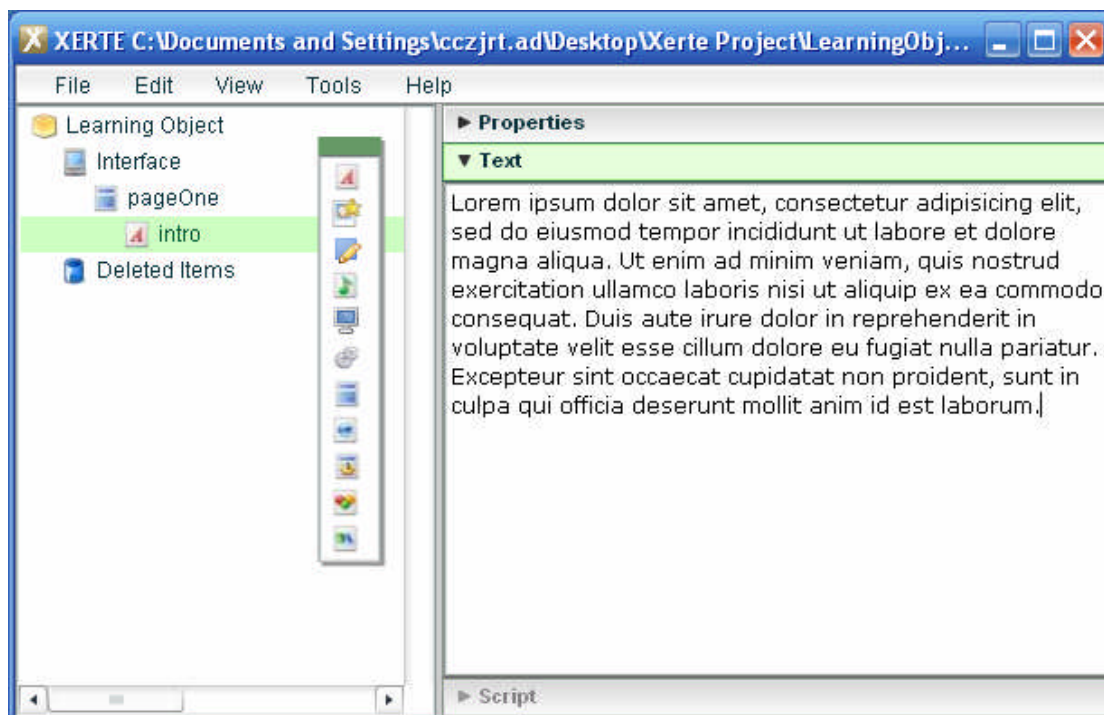
Pages are used to organise content. If you are using the default interface, the *title* property will be shown in the interface. You will add content and interactivity to pages. Pages provide structure in your piece.



The default interface provides a simple way of allowing users to navigate through a sequence of pages. As you develop a piece using the default interface, you will add pages to the interface, and add content to the pages.

Text Icon

Use text icons to add text to pages, and control the position, layout and appearance of the text.

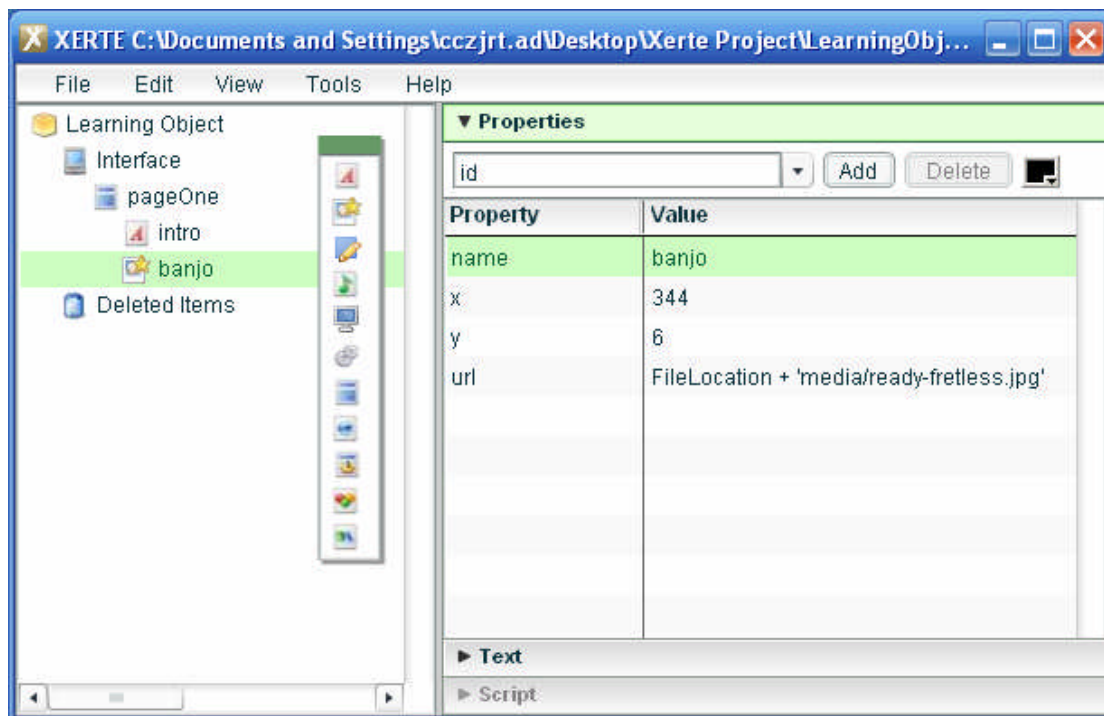




You can format text with html tags. The HTML tags supported are `<a>`, ``, `
`, ``, `<i>`, ``, `<p>`, ``, `<u>`. You can also specify a style sheet for the learning object on the Learning Object icon's properties.

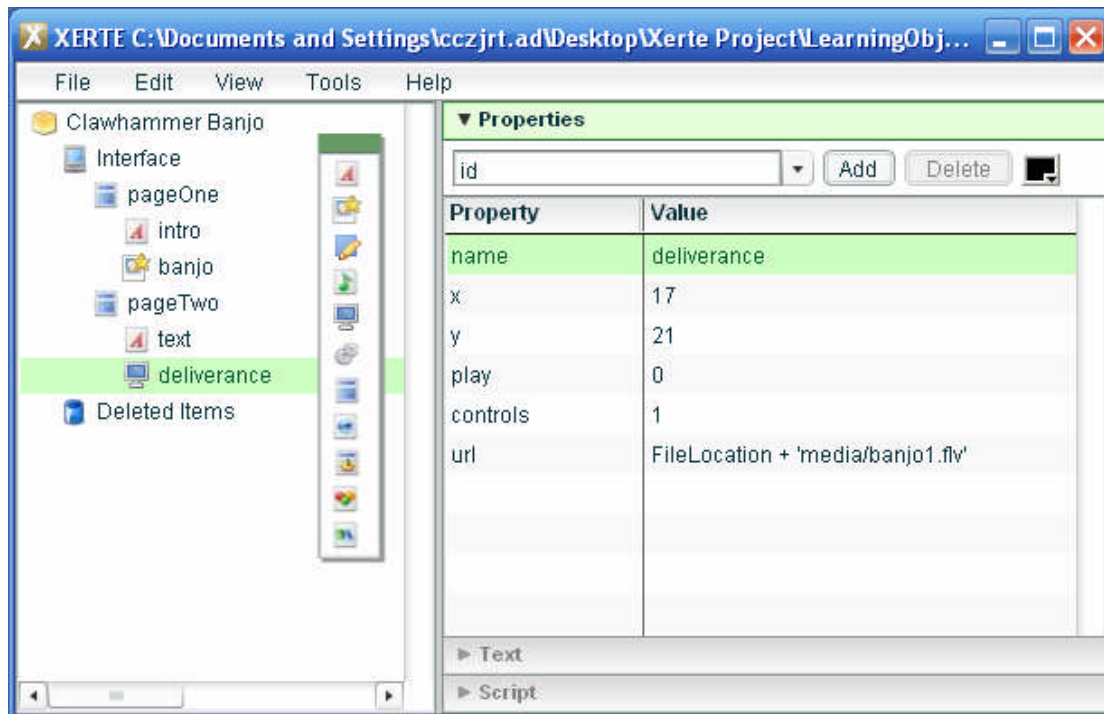
Graphic Icon

The graphic icon allows you to add still images and flash animations to your learning object.



Movie Icon

Use the movie icon to add FLV movies to your learning object.

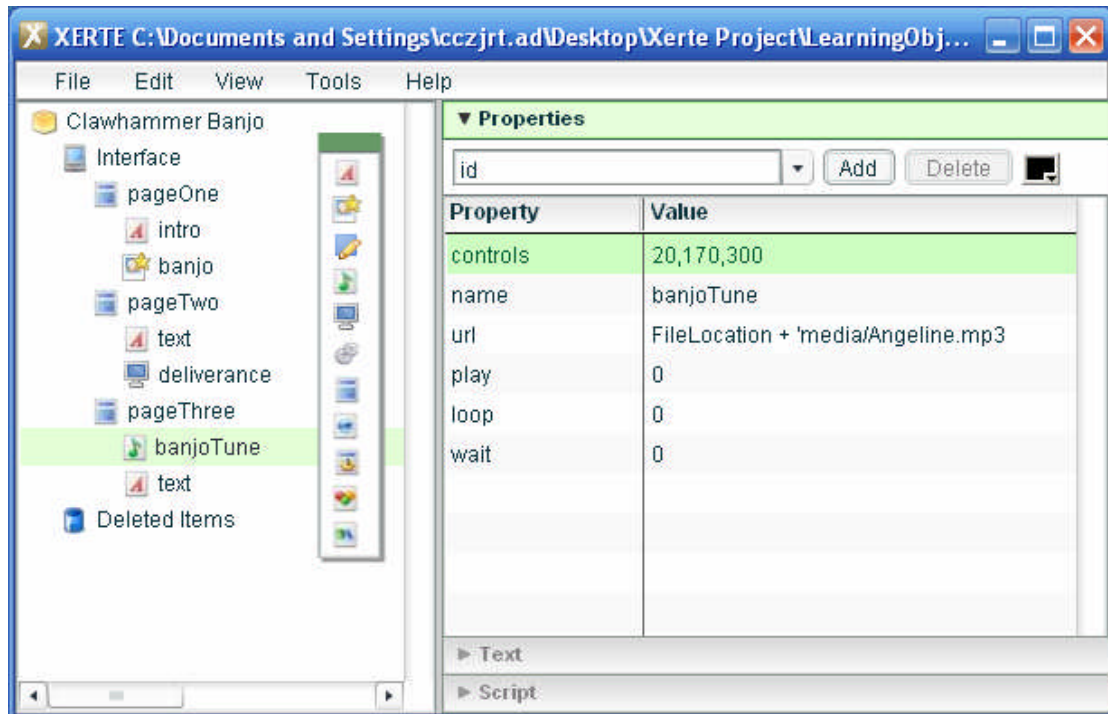


Sound Icon

Use a sound icon to add sound to your piece. Sounds can be effects, narration or music. You can choose to add controllers for the user to control the sound,

choose to loop the sound and determine whether the flow waits for the sound to finish before moving on to the next icon in the document tree.

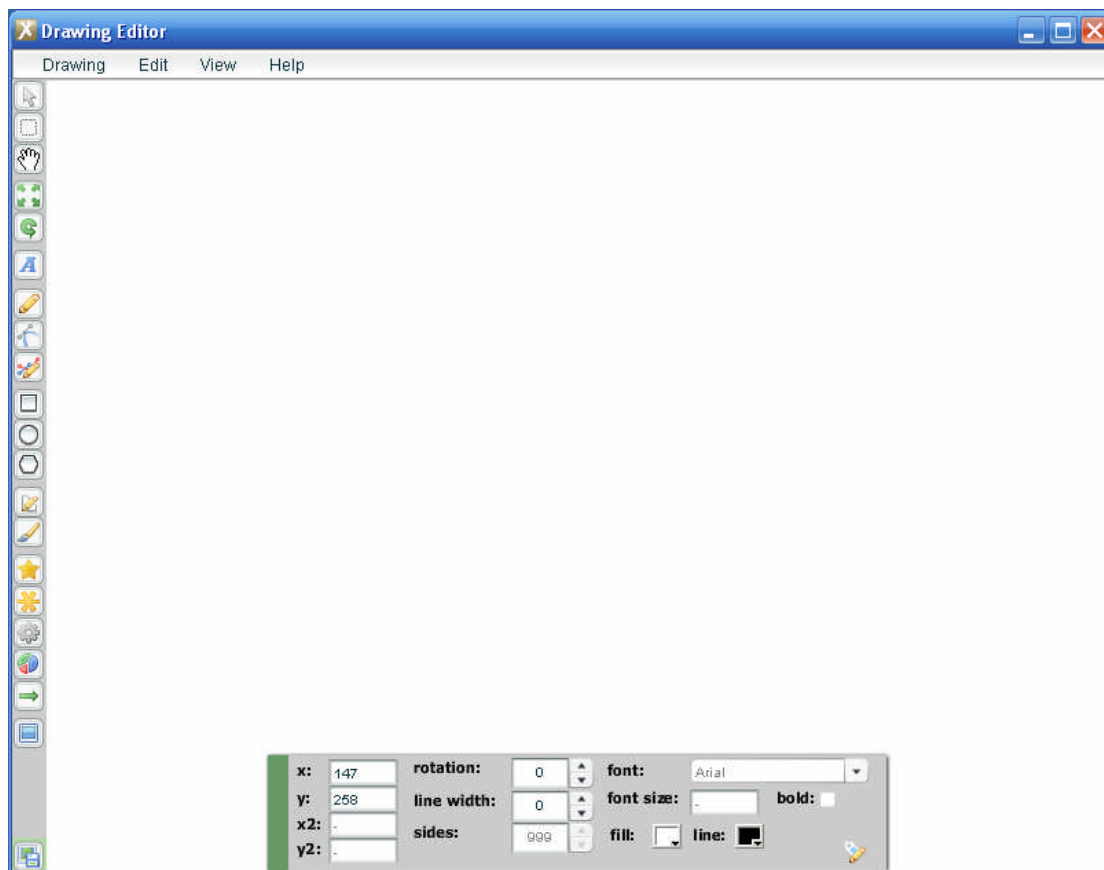
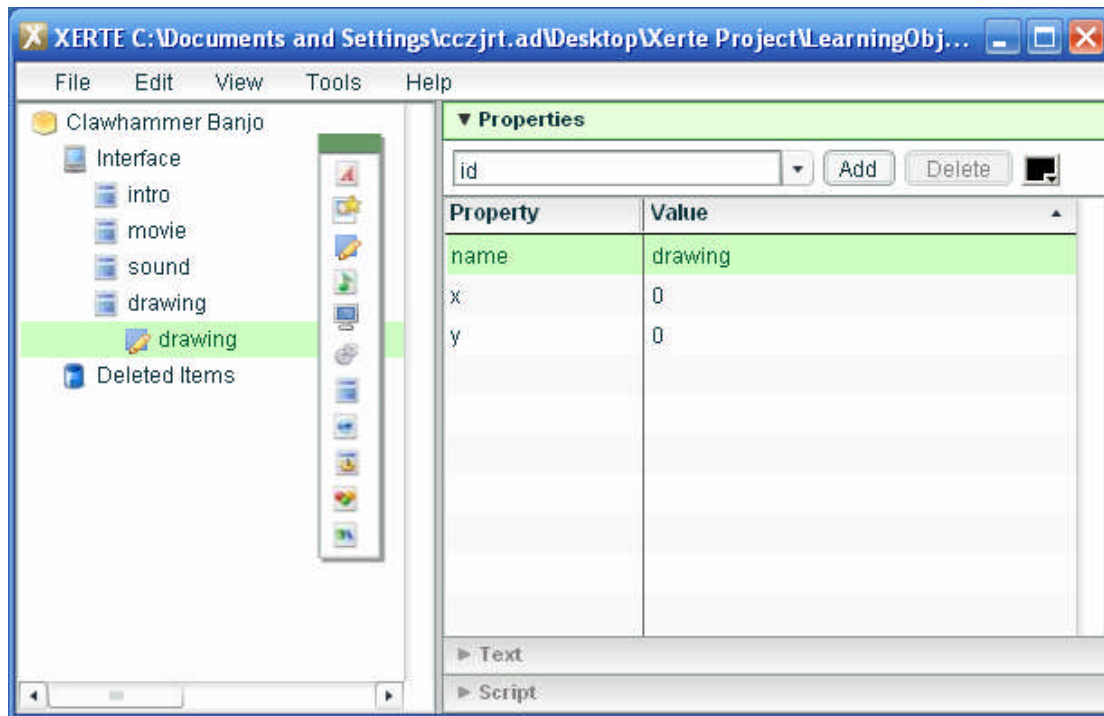
You can synch content to the playback of the sound by adding pages to the sound icon, setting the synchPoint property of the pages and adding content to them.





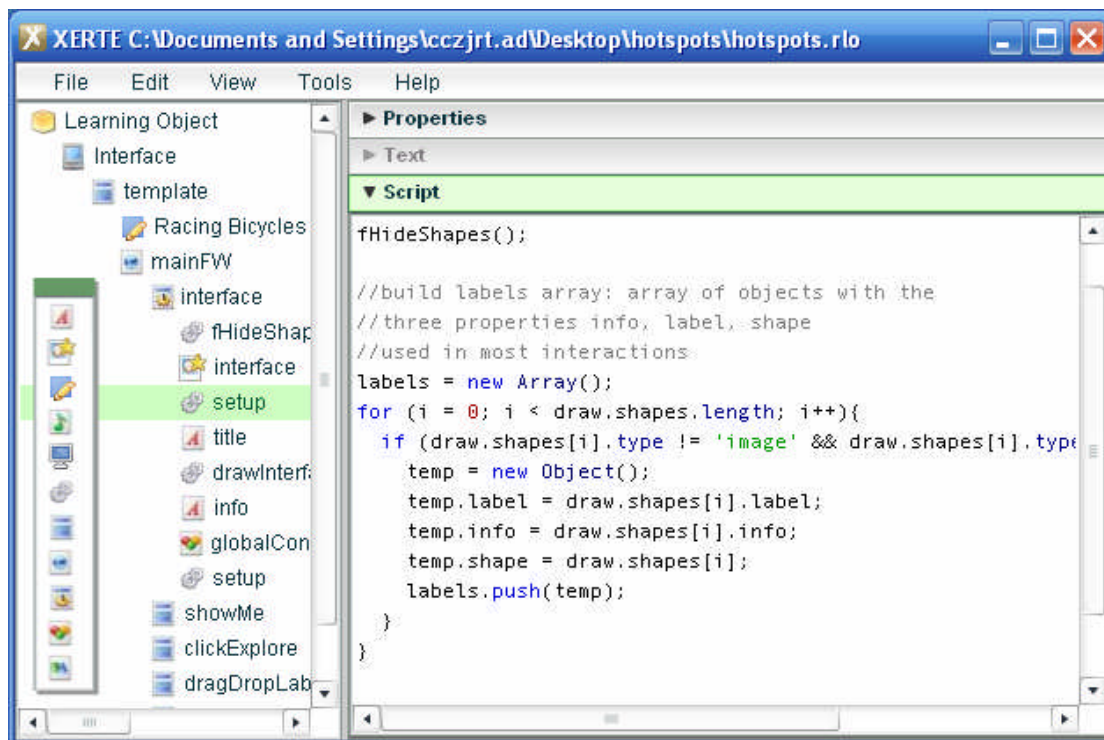
Drawing Icon

Use the drawing icon to add your own drawings to the learning object. Add a drawing icon and double click it to open the drawing editor. You can use the drawing object to design interface elements, create a rough storyboard before development begins, or create simple diagrams and illustrations.



Script Icons

Use script icons to add code to your learning objects. When script icons are encountered in the flow, the code they contain is executed.



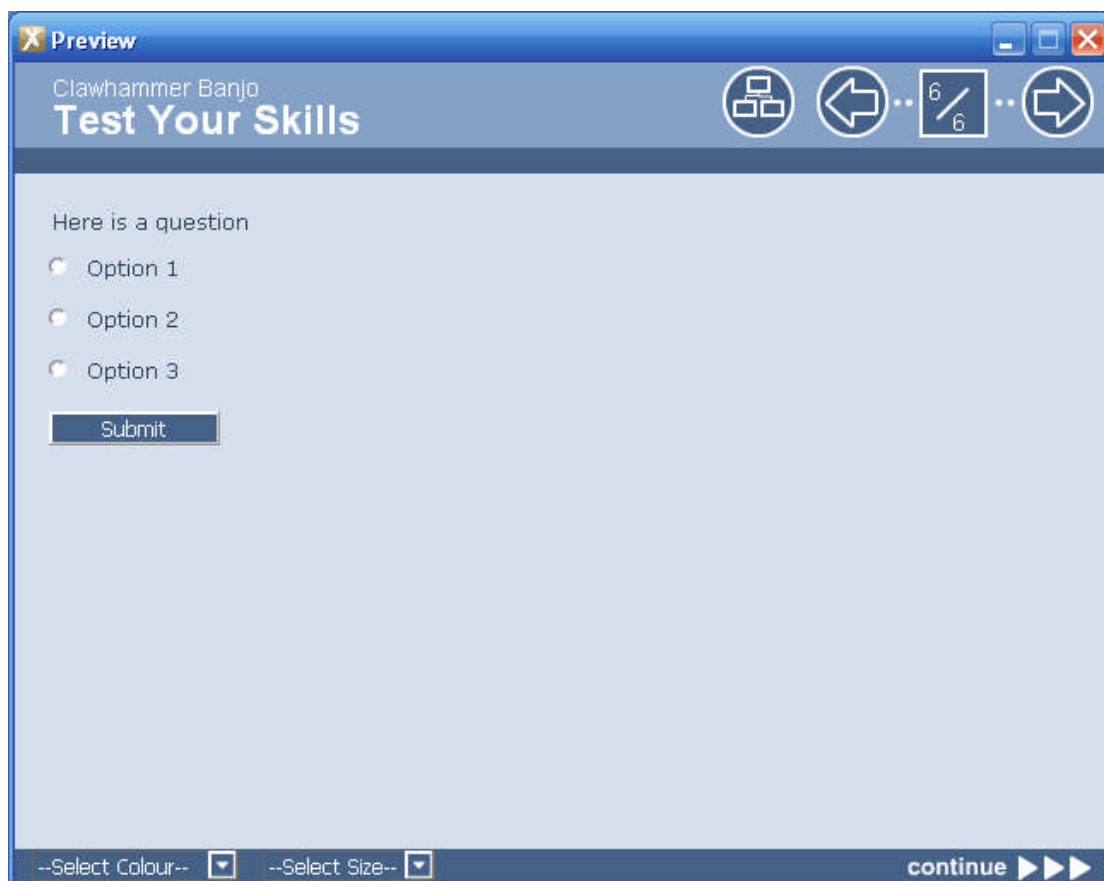
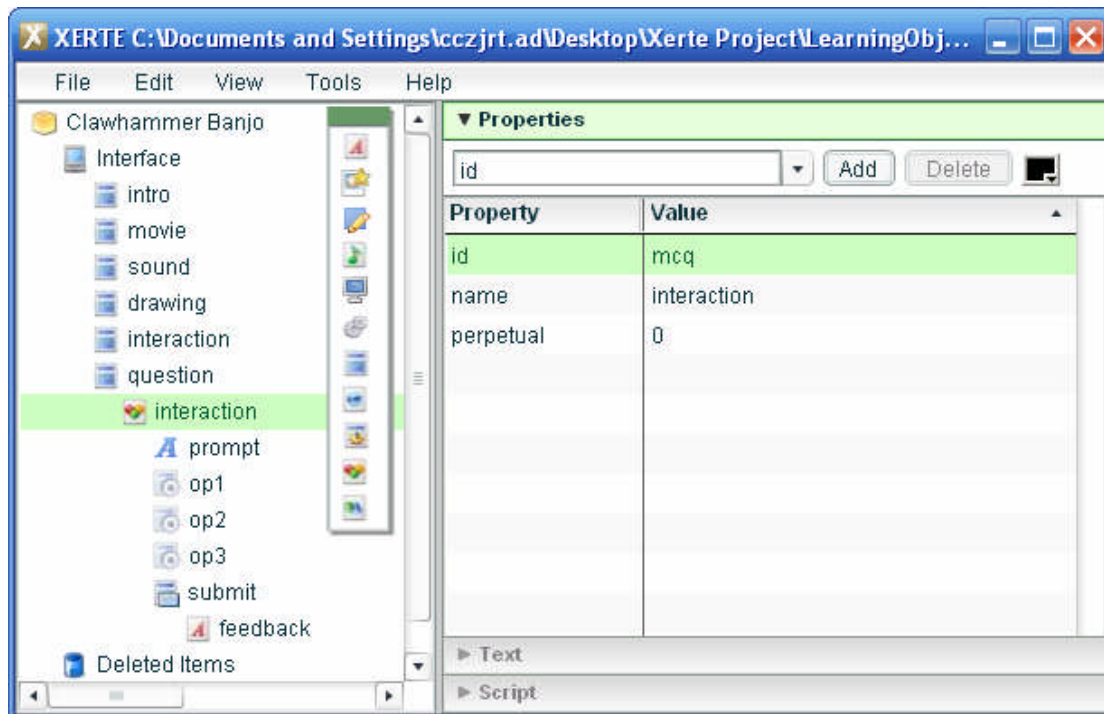
If you already write some ActionScript / JavaScript, the syntax will be familiar to you. There are a few things to bear in mind:

- The language is ActionScript 1.0.
- The language has been extended with additional classes, and so Xerte's icons have their own properties, methods and events that you can script.
- There is no **var** keyword. All variables are executed in the same scope.
- There is no **function** keyword. You can usefully create script functions using script icons, but they are not true functions, and are more akin to subroutines.
- If you require programming functionality not supported by Xerte (such as recursion) you can write functions in Flash, compile the swf, and load that swf into Xerte from where you can call the functions it contains, passing in values from the learning object.

See the help file for more information on Xerte's scripting capability.

Interaction Icons

Interaction icons manage interactivity in your learning objects. You attach 'responses' to an interaction icon to create interactive elements on the page. The 'response' is a page of content that will be triggered when a particular action occurs, such as a user pressing a button, or drag and dropping an item into a target area.



You can add whatever content you like to the response including further interactions. This presents opportunities for easily creating complex interactivity very quickly.

You can choose to set up interactions to halt the flow until the interaction exits.

You can add many responses to an interaction, and they can all be different types of responses.

The *perpetual* property of an interaction icon determines whether the flow will halt when the interaction is encountered or not. If set to 0 (false) then the flow halts when the icon is encountered on the flow line. To move on past the interaction icon in the flow, you must create a response and set its *exit* property to 1. If the perpetual property is set to 1 (true), the interaction is set up as soon as the flow encounters the icon, and the flow immediately moves on to the next icon in the document tree.

Interaction Responses

How Interaction Responses Work

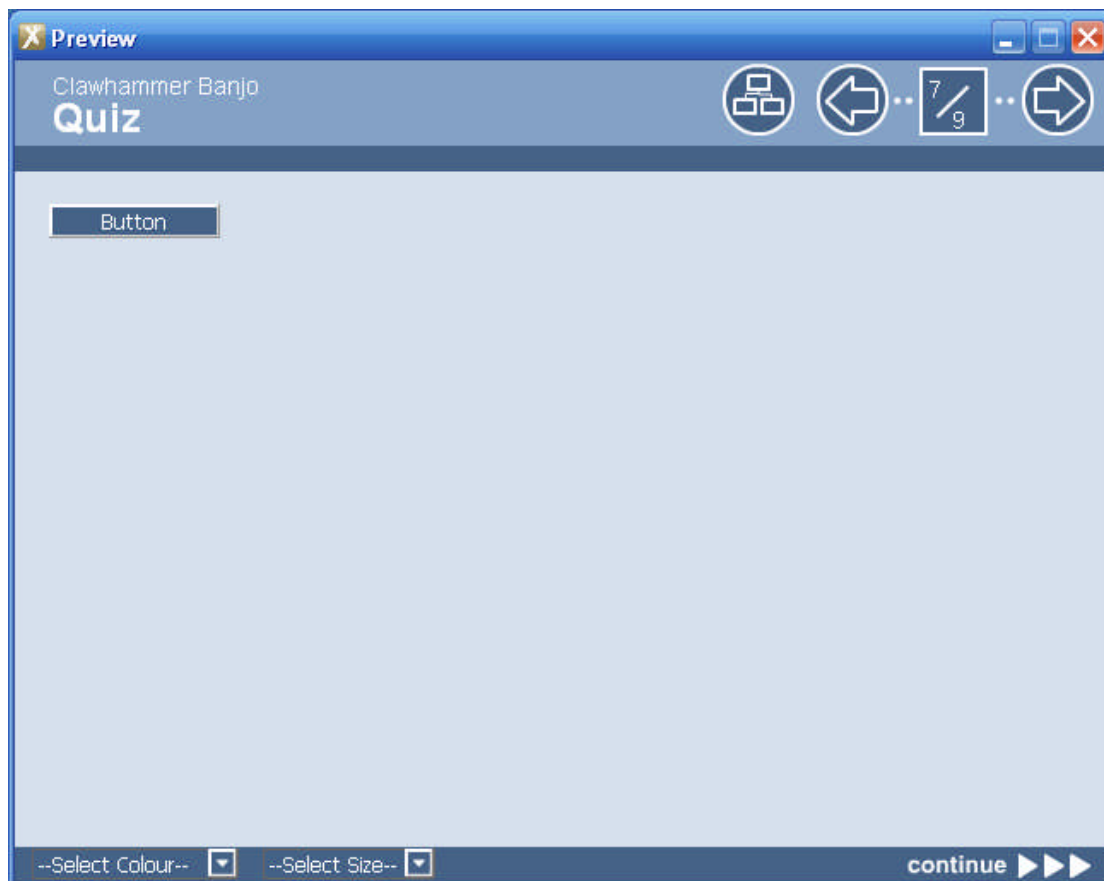
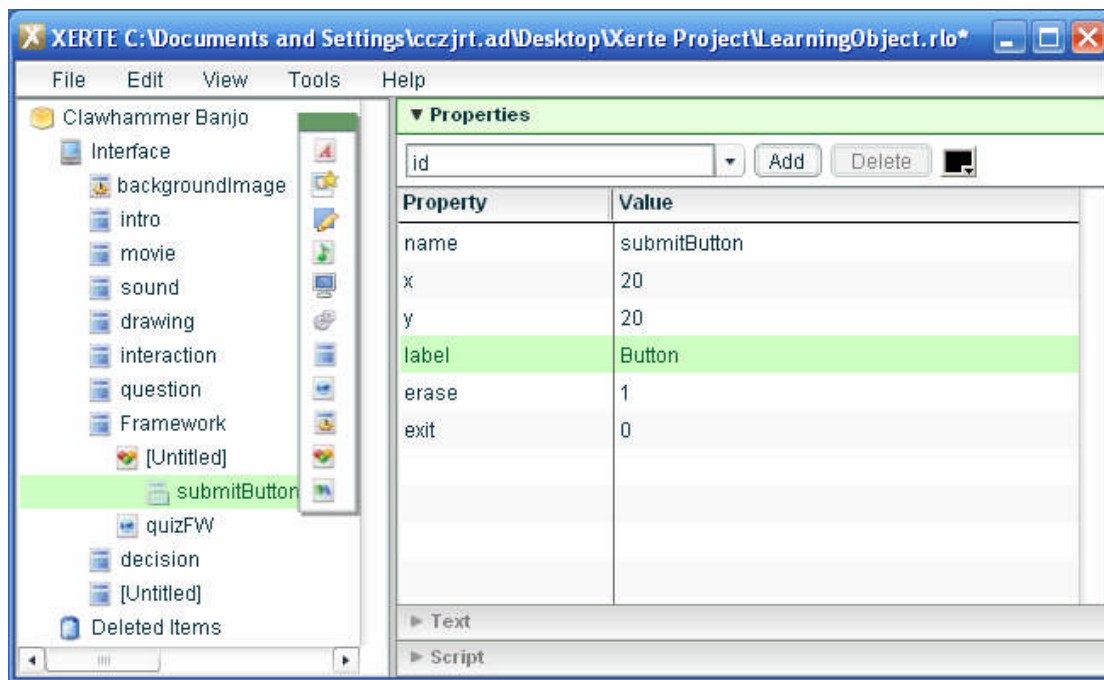
A response is two things: the interactive item, and a page of content to be shown when that item is interacted with. When the response is matched (a button is clicked, a checkbox checked or unchecked, or an item is rolled over, for example), its page of content is shown on the screen.

All interaction responses have two properties in common: *erase* and *exit*. If *erase* is set to 1 (true) then when this response is matched, all other responses of the interaction currently visible on the screen will be erased. If set to 0 (false) then other responses remain visible.

If the *exit* property is set to 1 (true) then the interaction will exit, and the flow will move on down the document tree – possibly to encounter a new interaction. If set to 0 (false) then the interaction remains visible.

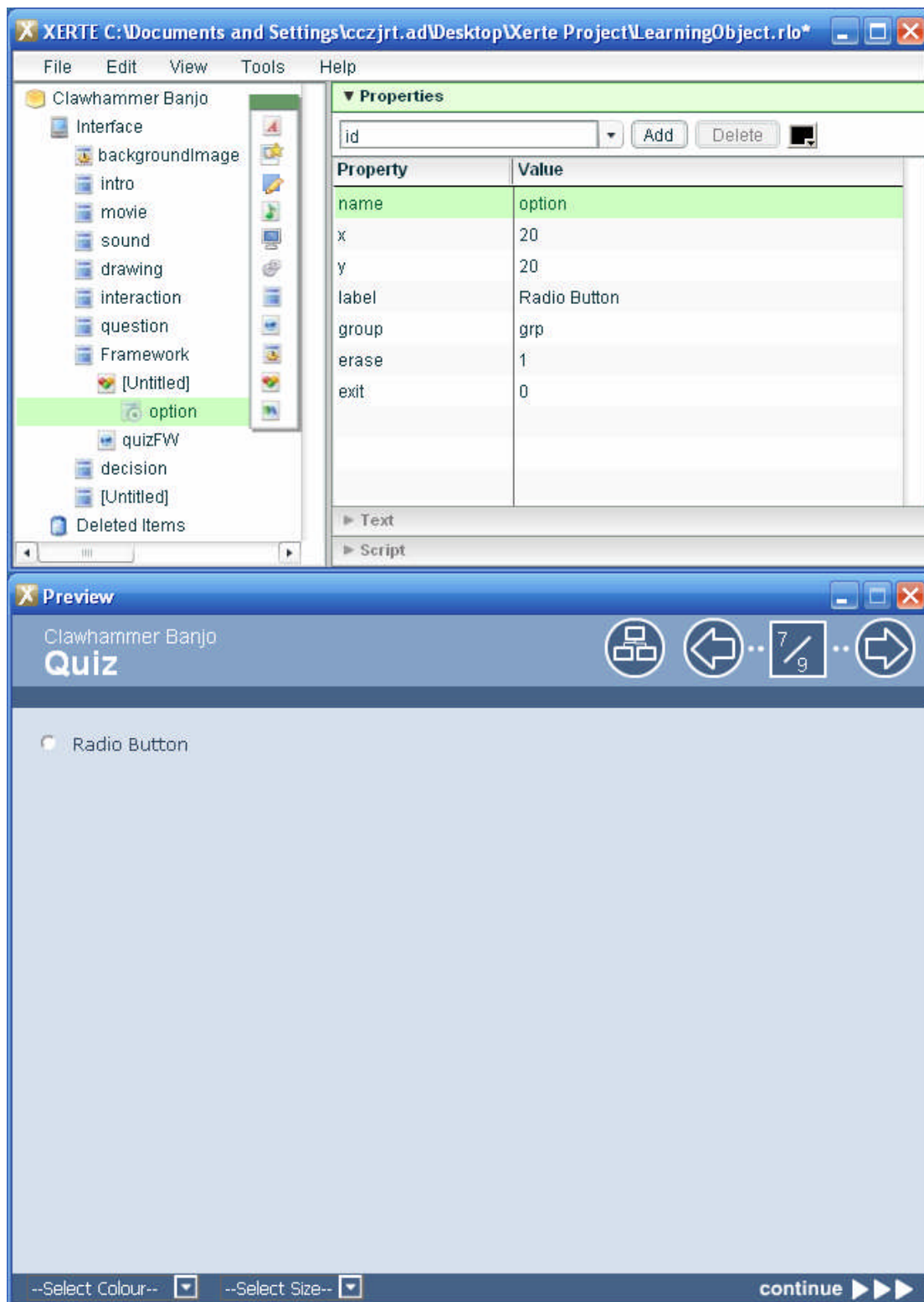
Buttons

Buttons allow you to create a button for the user to press using Xerte's default button, or to use an external graphic or swf as a button. If you base a swf on the button template in the templates folder, you can create eight-state buttons that can be set as checked / unchecked and enabled / disabled. You can build your own custom set of buttons to use in your learning objects.



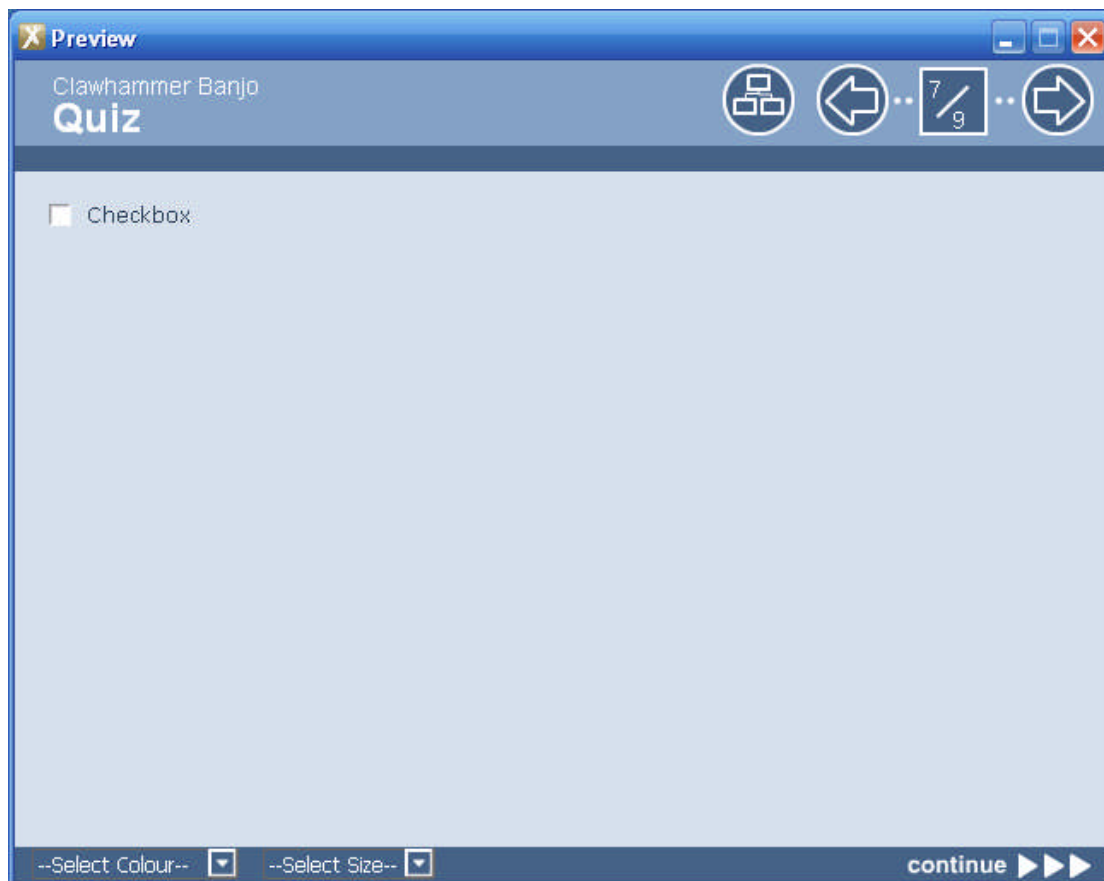
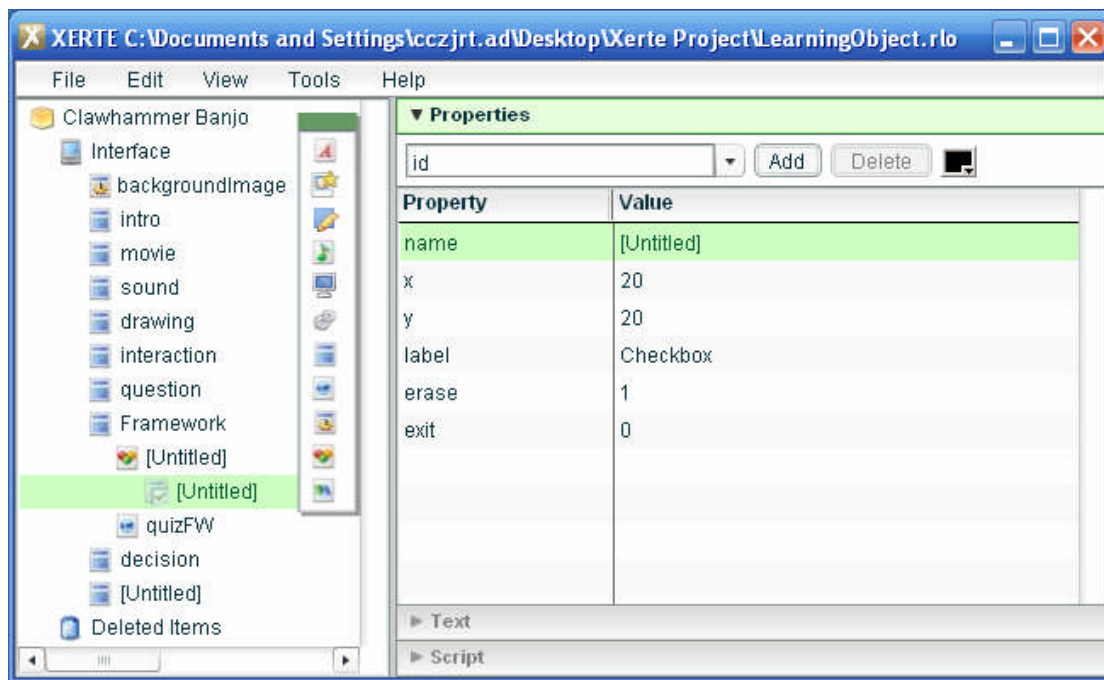
Radio Buttons

Radio buttons function exactly like a button response, presenting a radio button for the user to interact with.



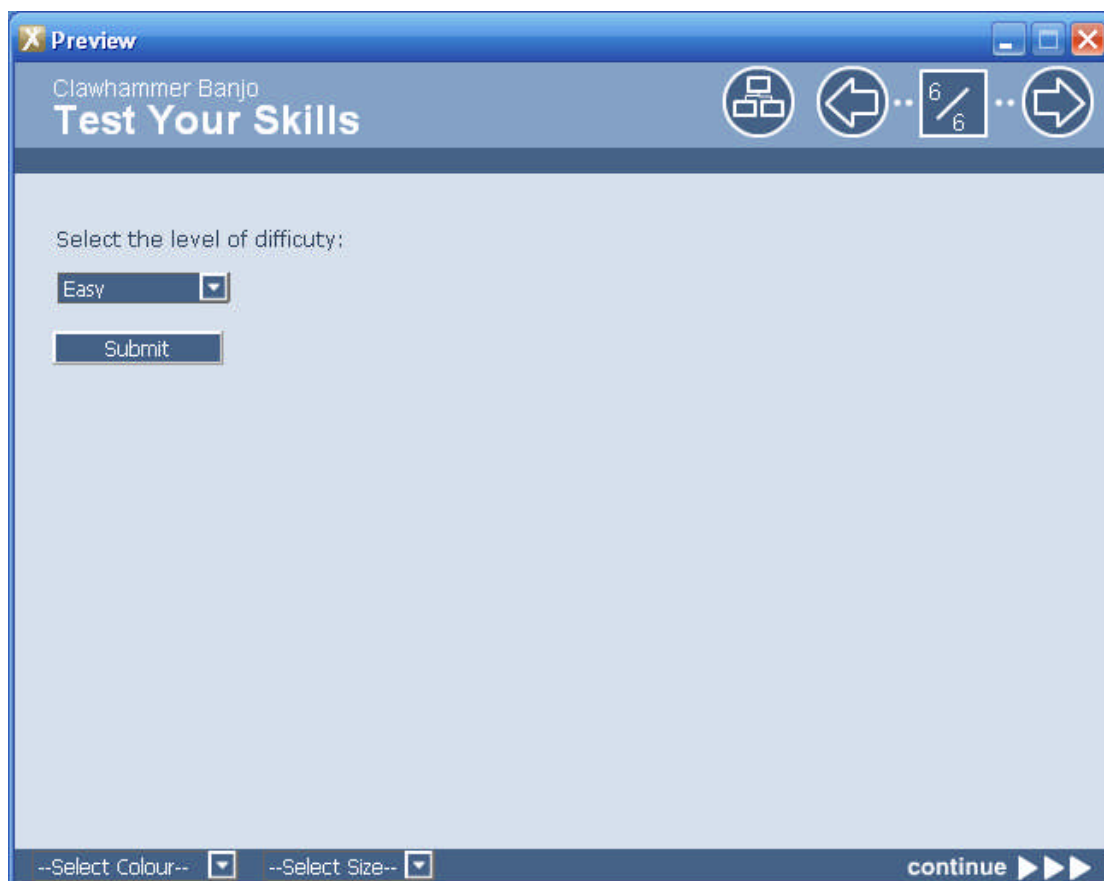
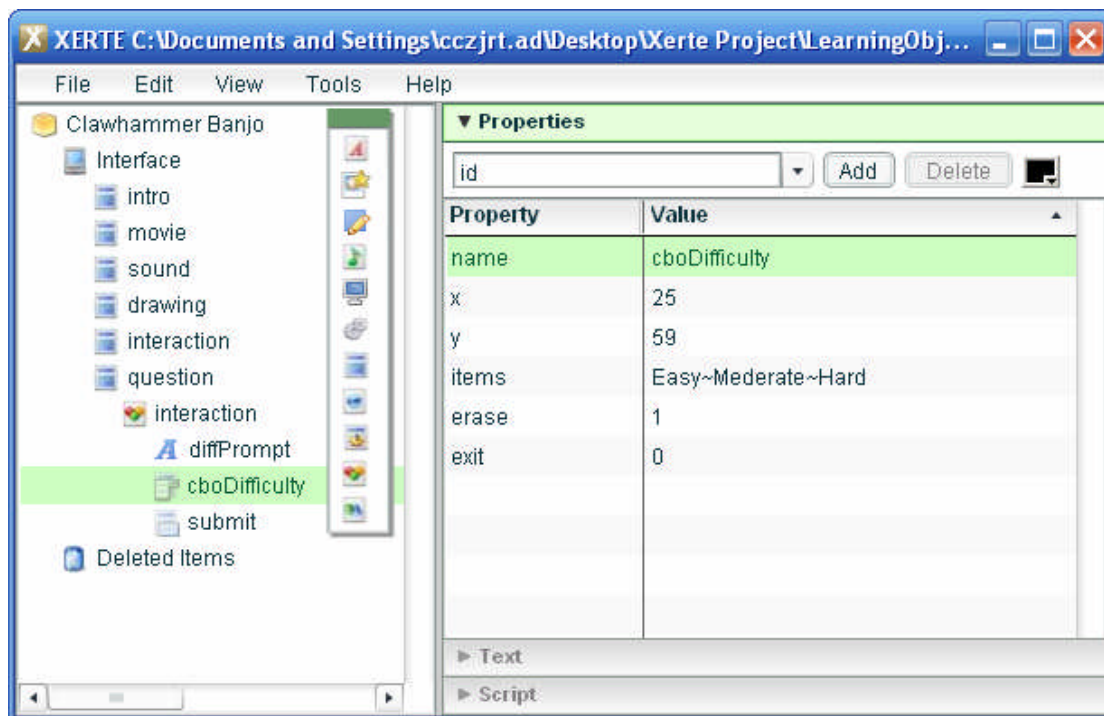
Checkboxes

Checkboxes function exactly like a button response, presenting a checkbox control for the user to interact with.



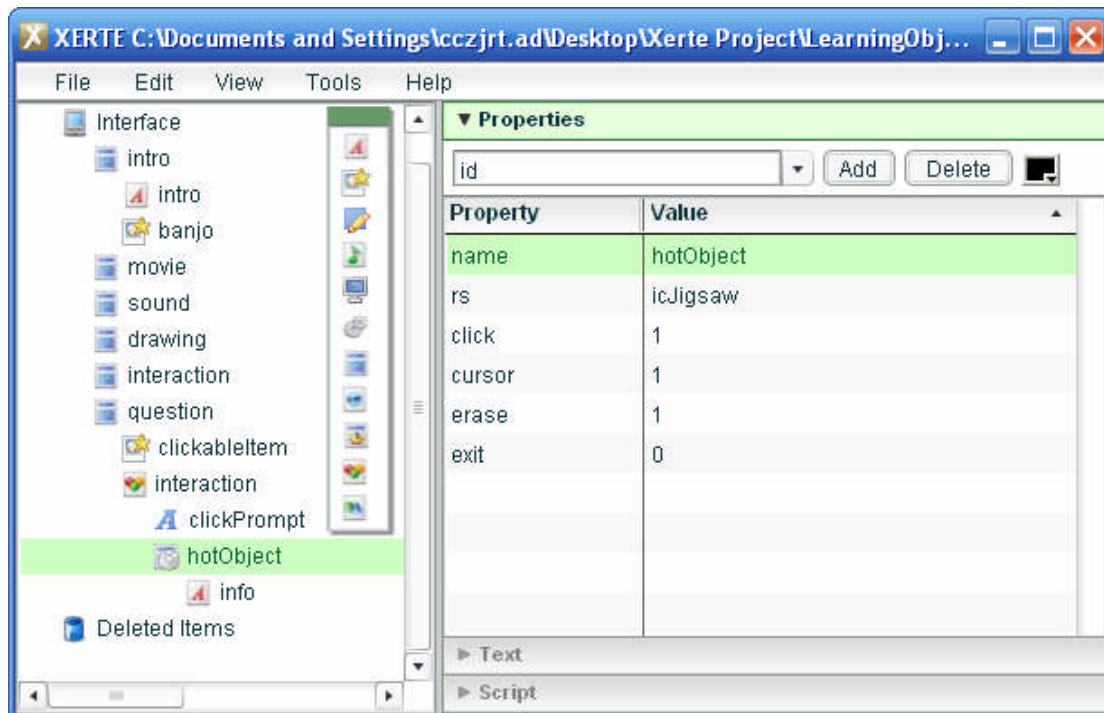
Combo Boxes

A combo box is a drop down list of items for the user to choose from.



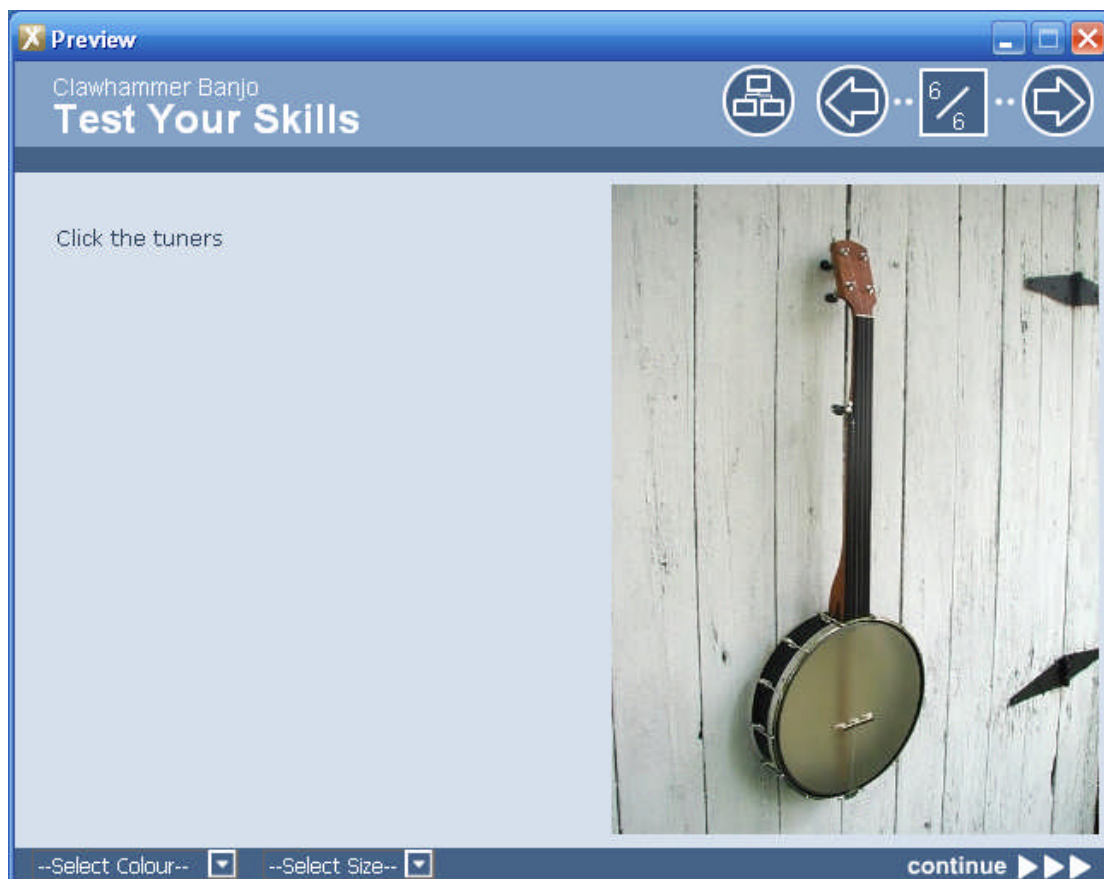
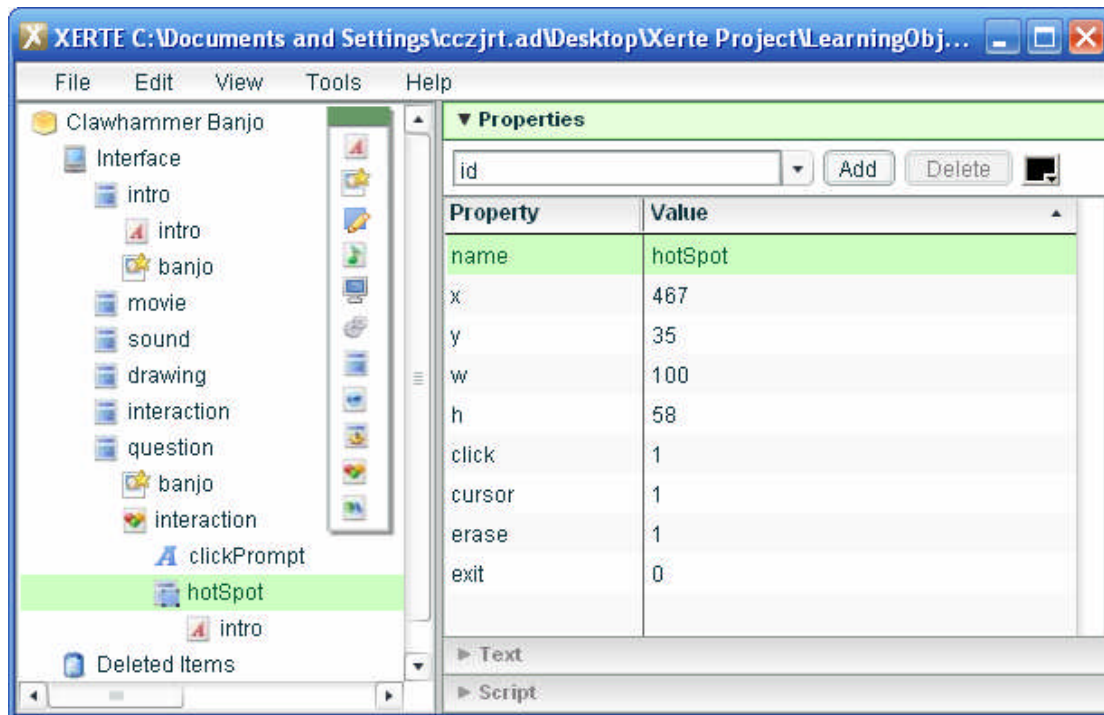
Hot Objects

Hot object responses allow an icon to become responsive to mouse events such as clicking or rolling over.



Hot Spots

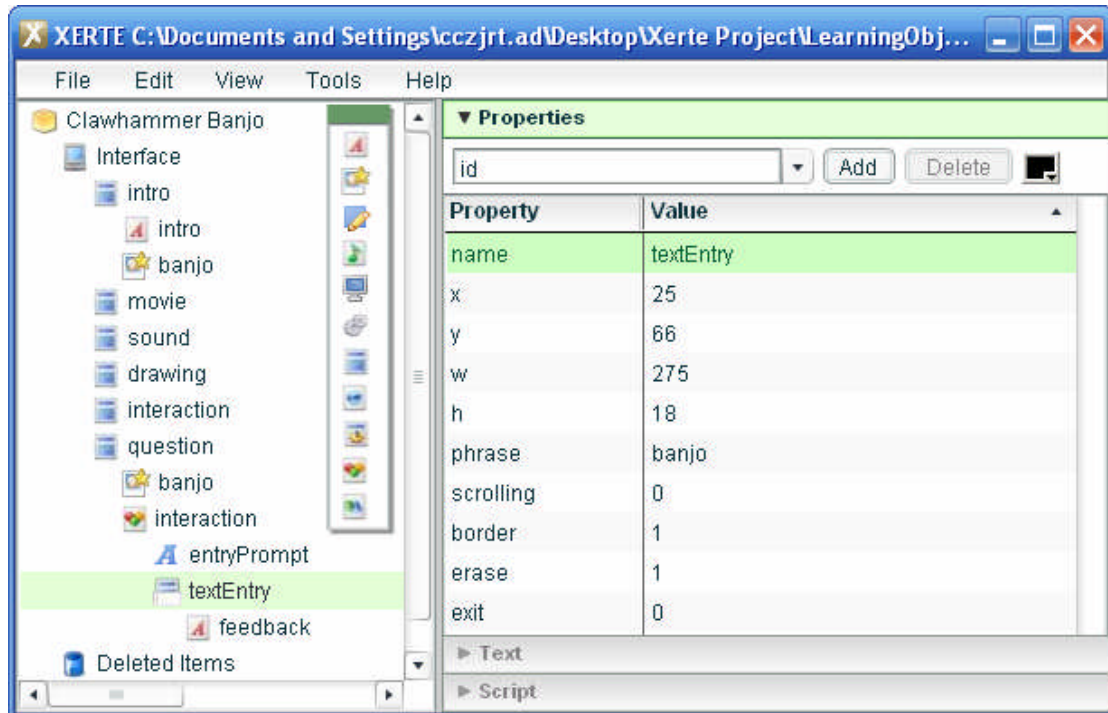
Hot spots are very similar to hot objects but define a region on the screen which can be clicked or rolled over to trigger a response. The obvious use is in interactive diagrams.

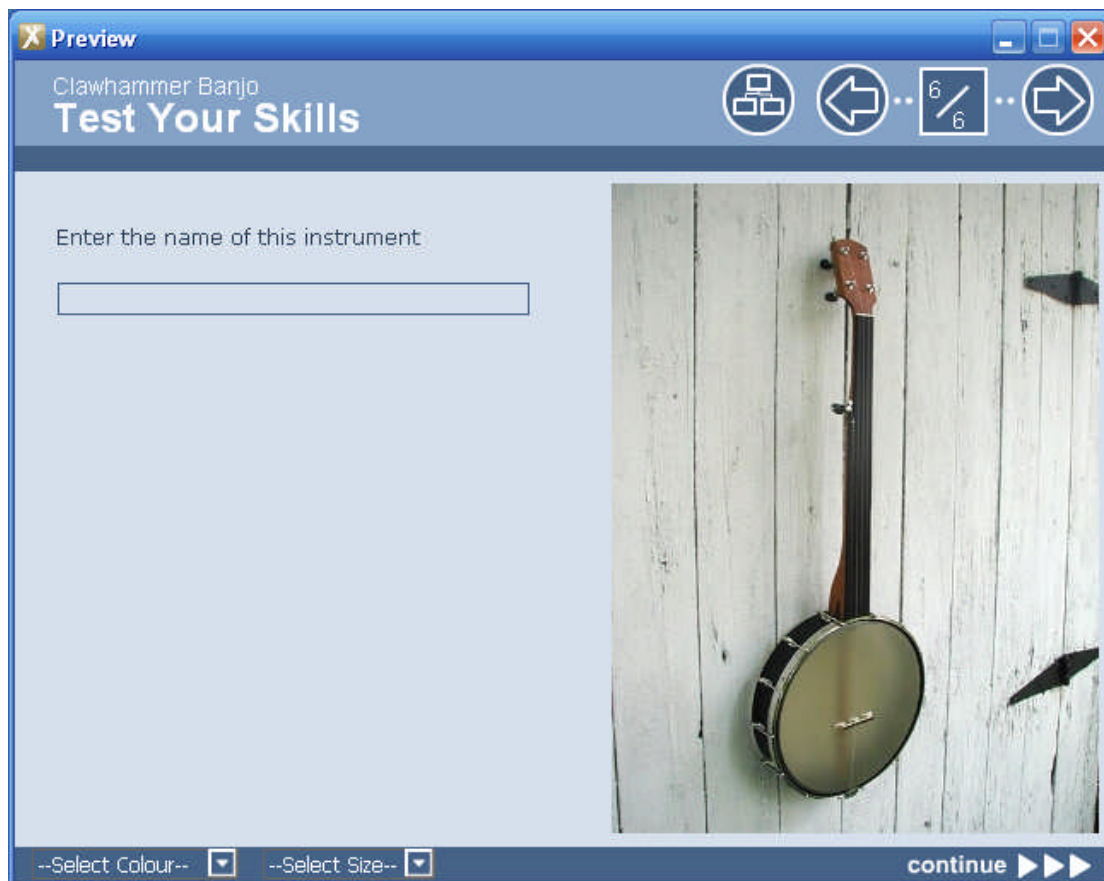


Text Entry

Allows the user to enter text. You can choose whether the response is triggered:

- When a given phrase is entered
- Any time the text in the entry box changes
- When the enter key is pressed

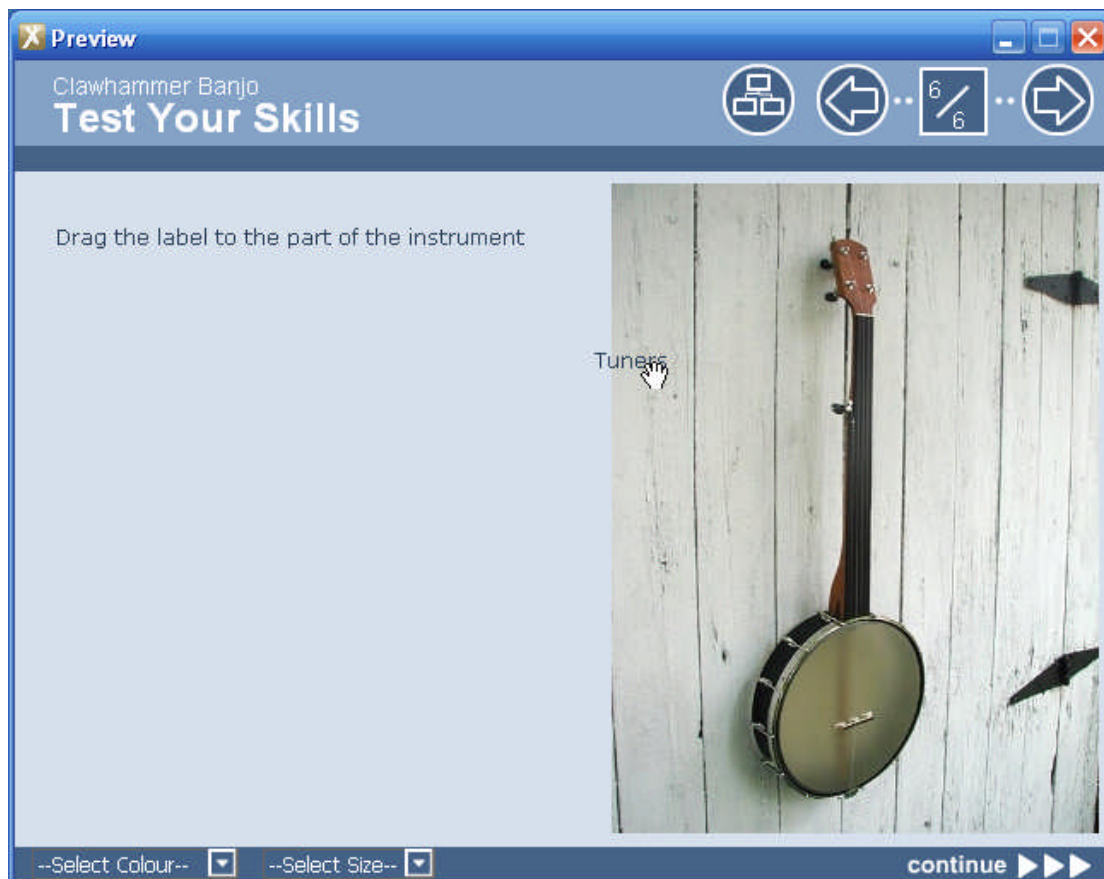
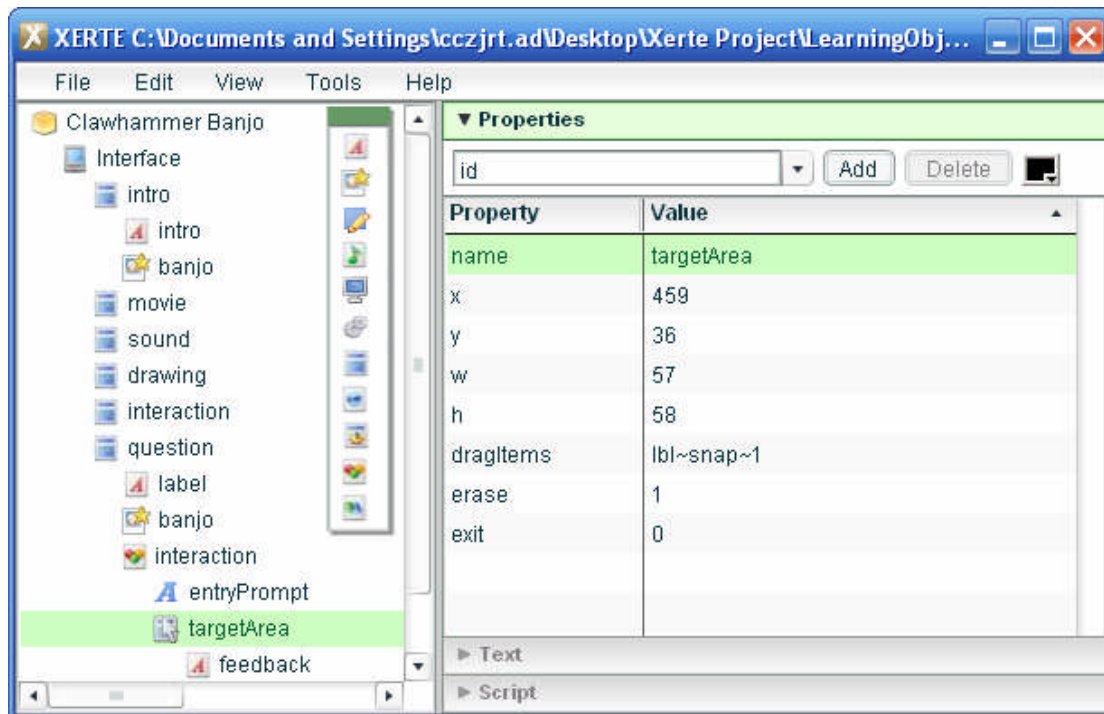




Target Areas

Target areas allow drag and drop interactivity to be set up. Drag and drop interactivity is the most complex to create. A target area defines a region on the screen that serves as the target area. Each target area has a list of items that, if dropped on it, will trigger its response. There can be several targets on the screen, and items can match more than one target. This allows drag and drop interactions of the one-one, many-one, one-many and many-many varieties to be created.

What happens when each item is dropped can also be determined – it can be snapped to the centre of the target area, or left; it can be allowed to be dragged again, or dragging can be turned off once it has matched a target area.



Time Limit

Triggers a response when a given period of time has elapsed.

Tries Limit

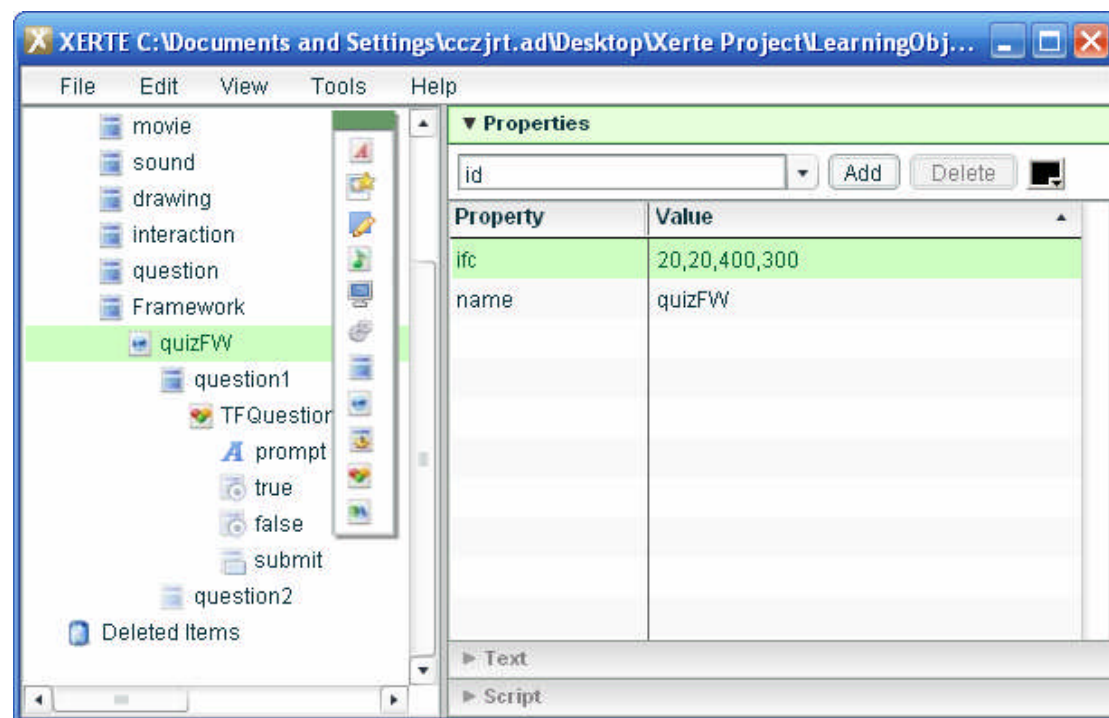
Triggers a response when an interaction's responses have been matched a given number of times.

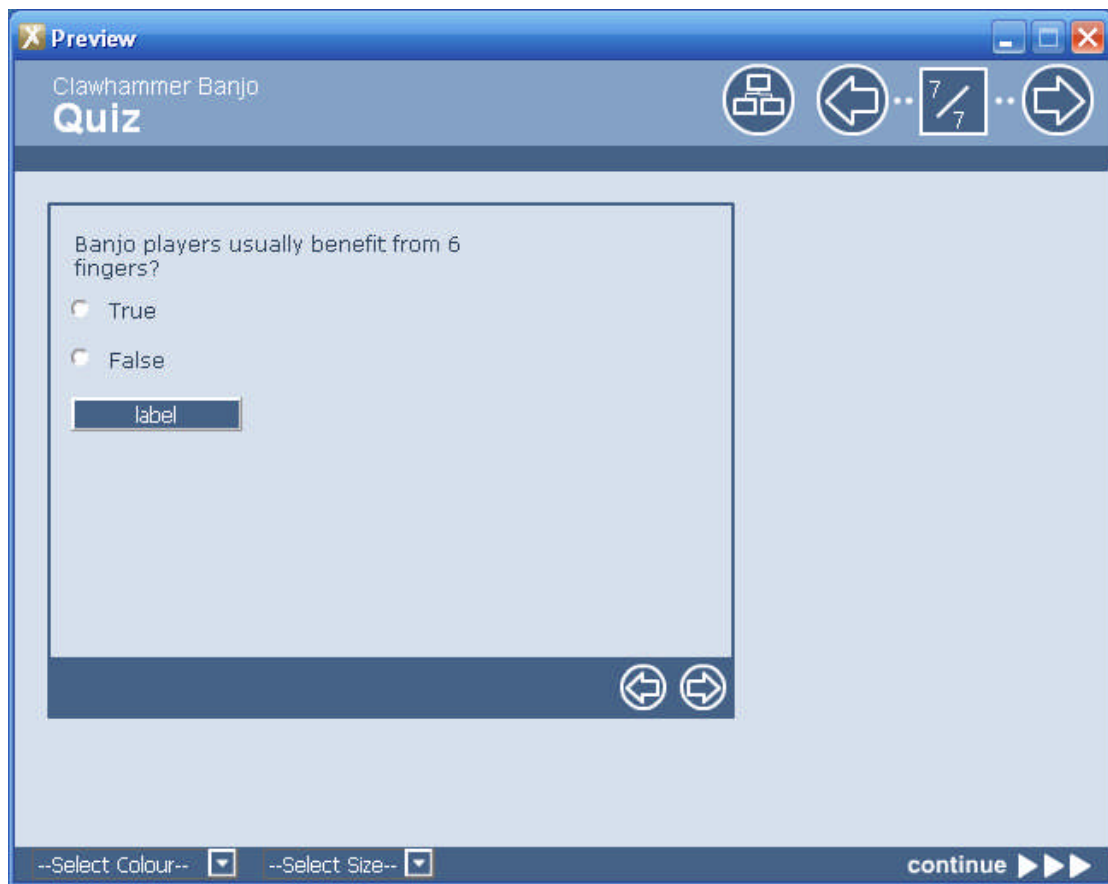
Events

Event responses allow you to listen to Flash's event system, and respond to those events using Xerte. This enables you to create very powerful interactivity, tapping in to Flash's native events. You can also broadcast and listen for your own events, but that's beyond the scope of this document...

Framework Icons

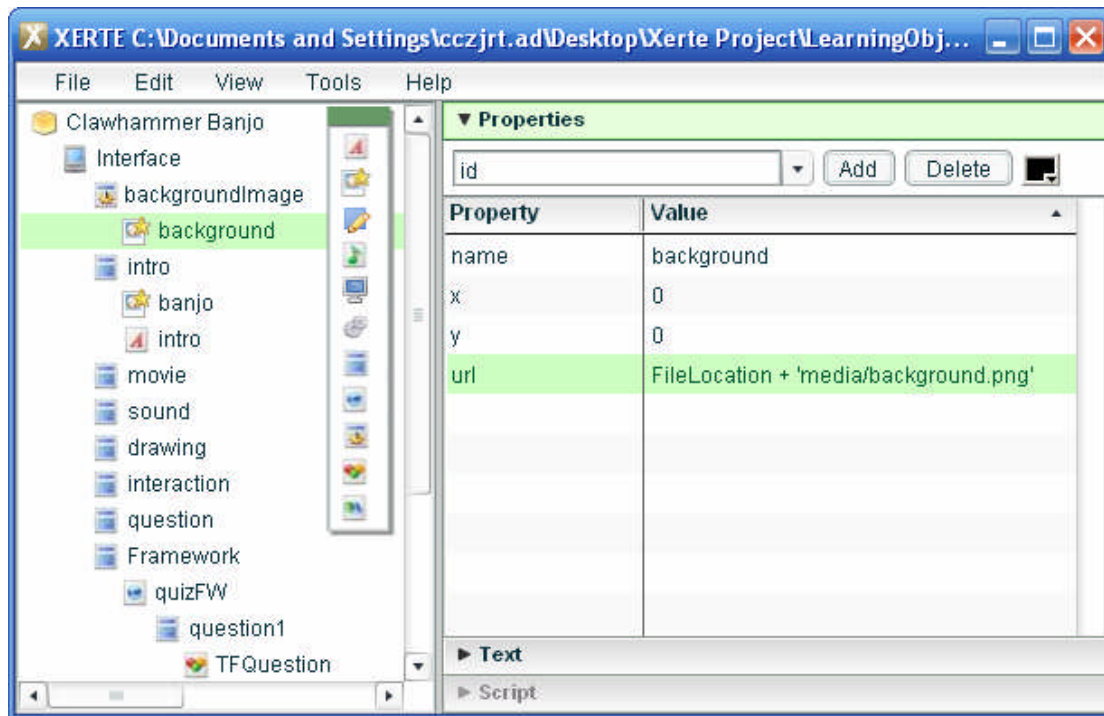
Framework icons allow you to create nested structures of pages. In fact, the interface icon is a type of framework. You add pages to a framework, and then create controls by which the user can navigate through the framework, or control the framework with code. You can use Xerte's default interface controllers to easily set up some simple controllers for a framework.





Framework Entry Frame

You can add a special type of page to a framework, called an 'entry frame'. The best way of describing the entry frame is a page that doesn't turn. It acts like a page, but its contents are always visible throughout the framework. It is a great place to put a background image for a learning object, controllers for a nested framework, or the elements of a custom interface.



Decision Icons

Decision icons play several roles in a project. They provide a means of sequencing content in several ways, and they are also helpful when pages contain many icons as a way to organise your file, and reduce the number of icons visible on the screen at any one time. One way of thinking of a decision icon is as a 'visual loop'.

Accessibility

We believe that accessible design improves the experience for everyone. In developing Xerte, we have sought to provide as many accessible features as possible, but there is no silver bullet. It is important to think about accessibility from the very beginning. For an overview of the accessibility features we have implemented, visit <http://www.nottingham.ac.uk/xerte> and follow the link to 'Accessibility' on the right hand side of the page.

Creating Models and Templates

Models

As you develop more and more content, you will begin to create more sophisticated pages and interactions that would be time consuming to rebuild every time you want something similar. One way that Xerte facilitates re-use is through the ability to save models. Models might be an entire learning object template, a well developed interaction or a particular page layout.

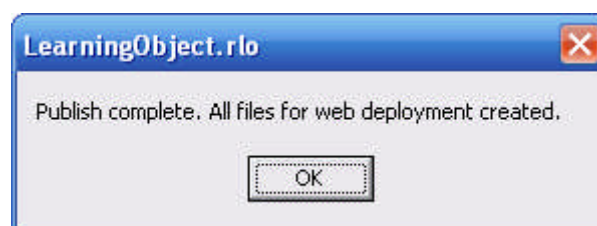
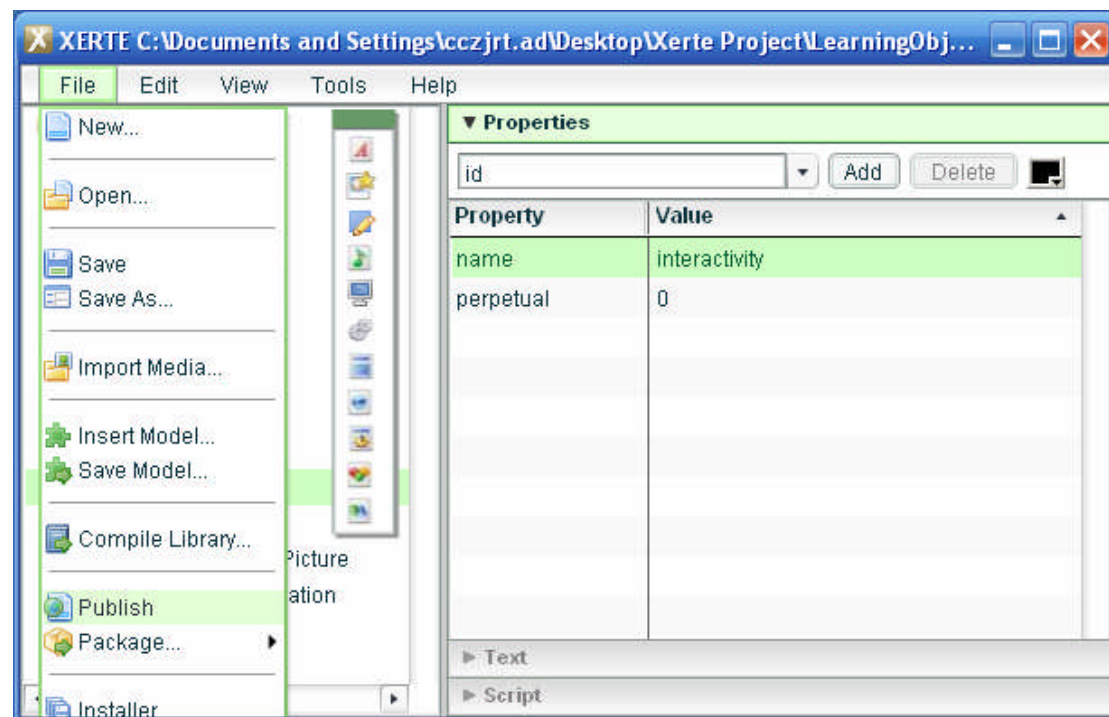
Any icon in Xerte can be saved as a model, allowing it to be easily re-used in another piece later. To save an icon as a model, select the icon you want to save, and select 'Save Model' from the file menu.

To insert a model you saved earlier, select the icon in the document tree select 'Insert Model' from the file menu and browse to the model you want to insert. You can develop a library of models that will greatly aid your productivity.

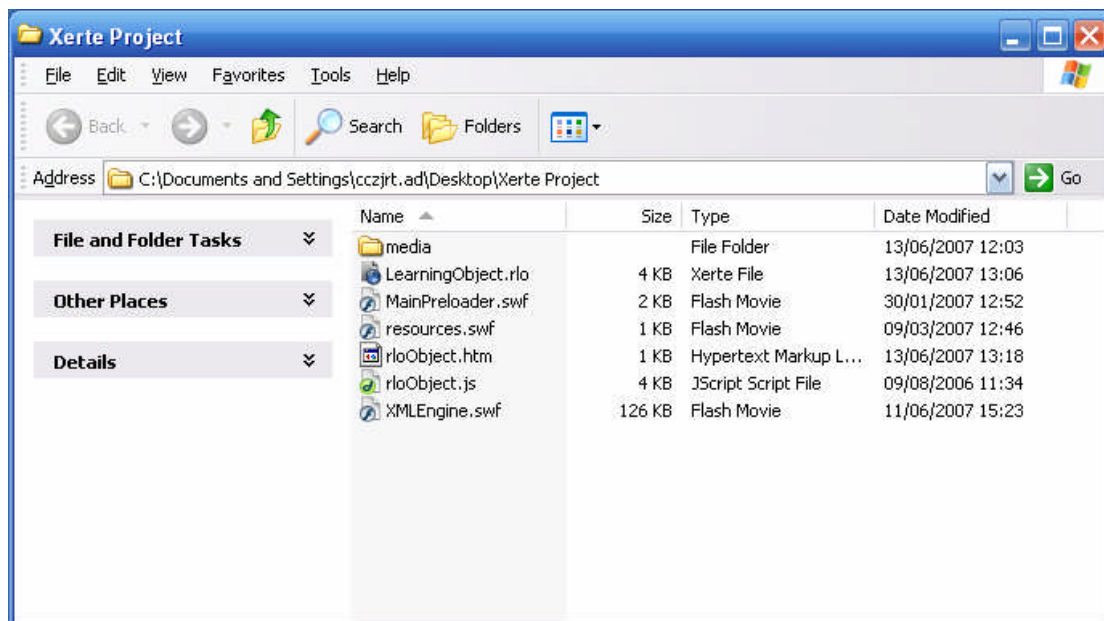
Deploying your Projects

Publishing a Project

To publish a project, simply select 'publish' from the file menu.



All the files you need to deploy to the web server are created for you. Now you should test the learning object and upload the files to a server, creating a link to rloObject.htm. You can change the name of rloObject.htm to something more suitable for your project.



Packaging a Project

To create a SCORM compliant content package, choose 'Package' from the file menu and then select the version of SCORM you want to comply with. Most LMS systems and VLEs support SCORM 1.2. You will be prompted to save the zip file. Xerte creates all the files required for the packaging, places them in the project folder, and then zips the project folder into the zip file. You can then upload that file into any SCORM-compliant system, such as WebCT, Dokeos or Moodle.