CGT 353 Lecture 4

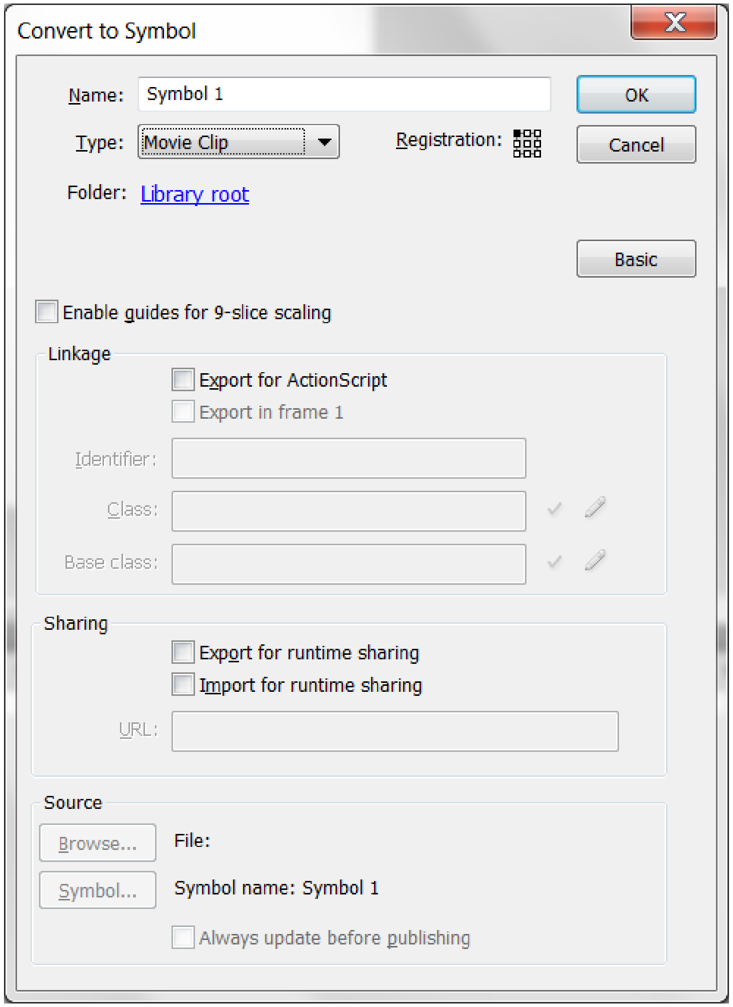
Symbols, Instances, and Libraries

Types and Purpose of Different Symbols

- Symbols are reusable components. They can be created by you or imported into Flash.
  - A Symbol can be reused throughout your movie or imported and used in other movies
  - A copy of a symbol used in the movie is called an **Instance**, which can have its own independent properties (like color, size, function, etc.) different from the original symbol
- Three types: **1) graphic, 2) button, and 3) movie clip**
  - **Graphic symbols are reusable static images that are used mainly to create animations**. Any drawn vector/plain text/imported bitmap (photo), or combinations of these, can be converted into a single controllable object: as a graphic symbol. They have only one frame in their timeline
  - **Button symbols are used for timeline navigation** - They add interactivity to the movie and respond to mouse clicks, key press or rollovers/rollout, and other actions. You define the graphics associated with various button states (Up/Over/Down/Hit), and then assign actions to the instance of a button. They have 4 frames in their timeline - one each for the up, over and down states, and one to define the hit area of the button
  - **Movieclip symbols are reusable pieces of flash animation -** consisting of one or more graphic/button symbols - thus they are flash movies within your flash movie. They have their own non-restricted Timeline (any number of layers and frames - just like the main timeline) that plays independent of the main movie's Timeline. The best thing about using movieclips is that you can control them from actionscript - you can change their dimensions, position, color, alpha, and other properties and can even duplicate and delete them
- Each basically differ in their behavior, which can change at any time in the Library…

Editing Symbols

- Double click a symbol to access its timeline and make changes to it.
- Can also use the symbol dropdown menu.
- Making changes in the symbol timeline will affect all instances



Convert to Symbol

Name: Symbol 1

Type: Movie Clip

Registration:

Folder: Library root

□ Enable guides for 9-slice scaling

Linkage

□ Export for ActionScript

□ Export in frame 1

Identifier:

Class:

Base class:

Sharing

□ Export for runtime sharing

□ Import for runtime sharing

URL:

Source

File:

Symbol name: Symbol 1

□ Always update before publishing

Registration of Symbols

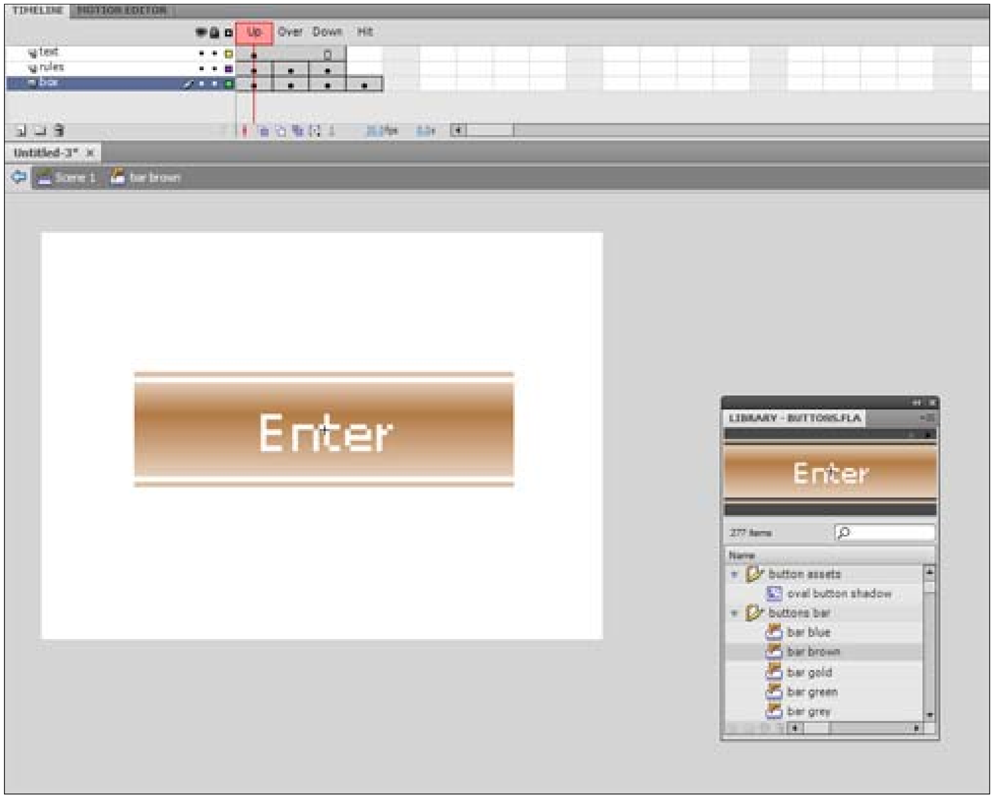
- The **registration point** of the main movie is the upper-left hand corner.
- You must physically move the content within a symbol to chance the registration point, which will affect all instances of the symbol…
- **Note:** The RP is different from the **origin** of a clip

Graphic Symbols

- Used primarily for static components….
- Graphic symbol timelines are synced to the main timeline.
- When the main timeline stops, the graphic timeline stops.
- Should not be used for clips that you need to loop.
- **Note:** Graphic symbols should be used sparingly. Movie clips are usually preferable.

Button Symbols

- A symbol that behaves like a push-button.
- Contains special frame for up, over, down, and hit states.
- **Warning:** Do NOT use dynamic or input text fields in your buttons

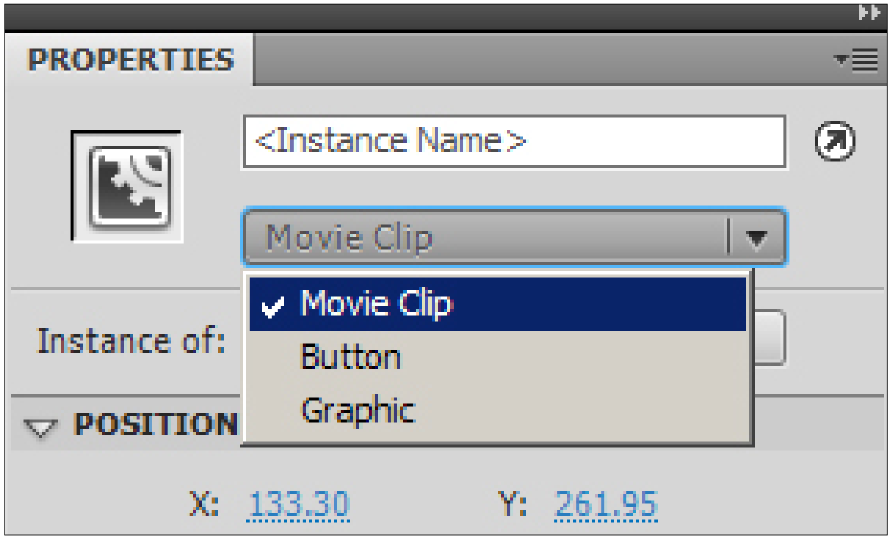


Movie Clip Symbols

- Movie clip timelines keep playing regardless of the main timeline.
- Can include all other times of symbols within them.
- Should use when you want a clip to keep playing.
- Need to either **a) include a stop() action inside it** or **b) directly tell the individual instance to stop via ActionScript**

Redefining Symbols

- Can change a symbol's behavior in the property panel, effectively changing the type of symbol.
- This <u>will not change the behavior of the symbol in the library</u>, which can be problematic.
- Can also break apart a symbol instance, which will not affect the main library symbol or any other instances.



<Instance Name>

Movie Clip

Instance of:

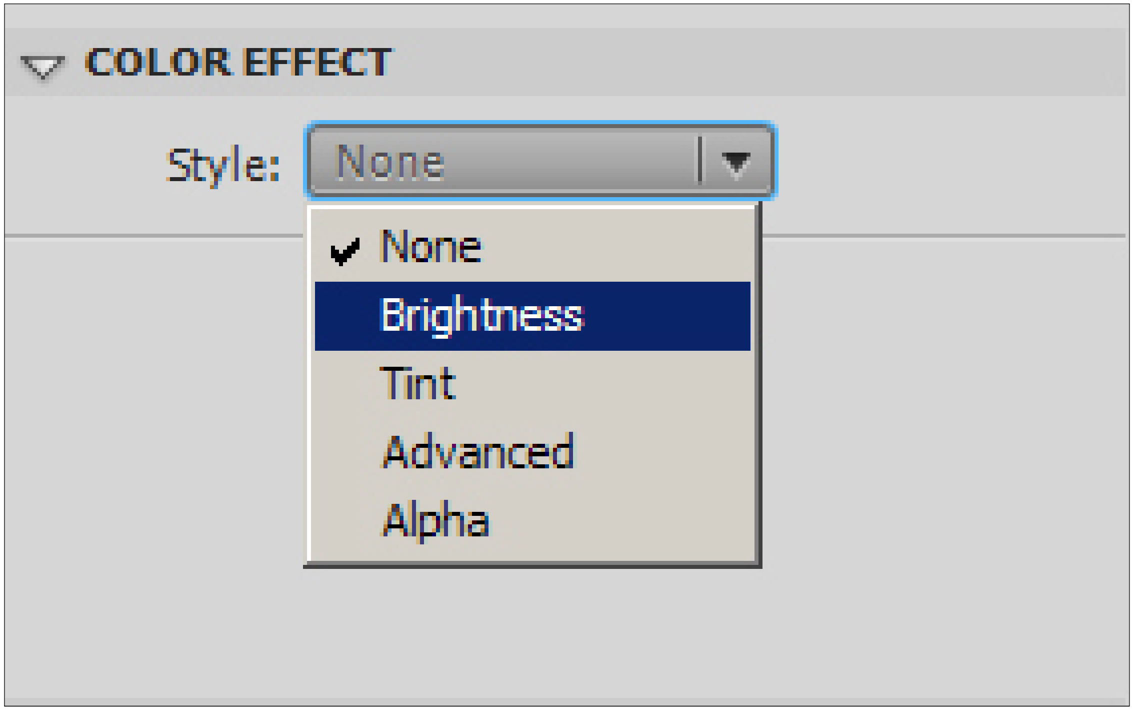
X: 133.30 Y: 261.95

Swapping Symbols

- Can swap different symbols while still retaining the properties of a particular instance

The Color of Symbols

- Allows you to apply a particular effect to a symbol, including:
  - Brightness
  - Tint
  - Alpha
  - Advanced



COLOR EFFECT

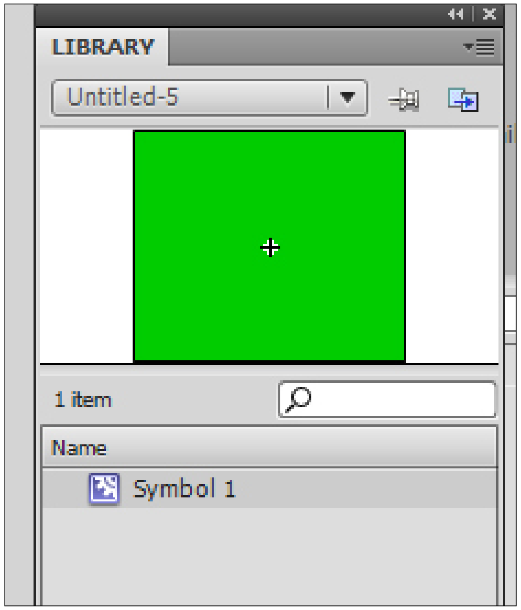
Style: None

Compound or Nested Symbols: Precedence Rules

- **IMPORTANT: Compound or Nested Symbols: Precedence Rules**
- Can recombine symbols to achieve different functionalities.
- Putting a button into a graphic symbol disables the button.
- Putting a movie clip into a graphic will allow the animation to play, but any buttons or sounds within the movie will be disabled.
- You can put a graphic or movie clip into any of the button states.
- You can put a graphic or button into a movie clip.

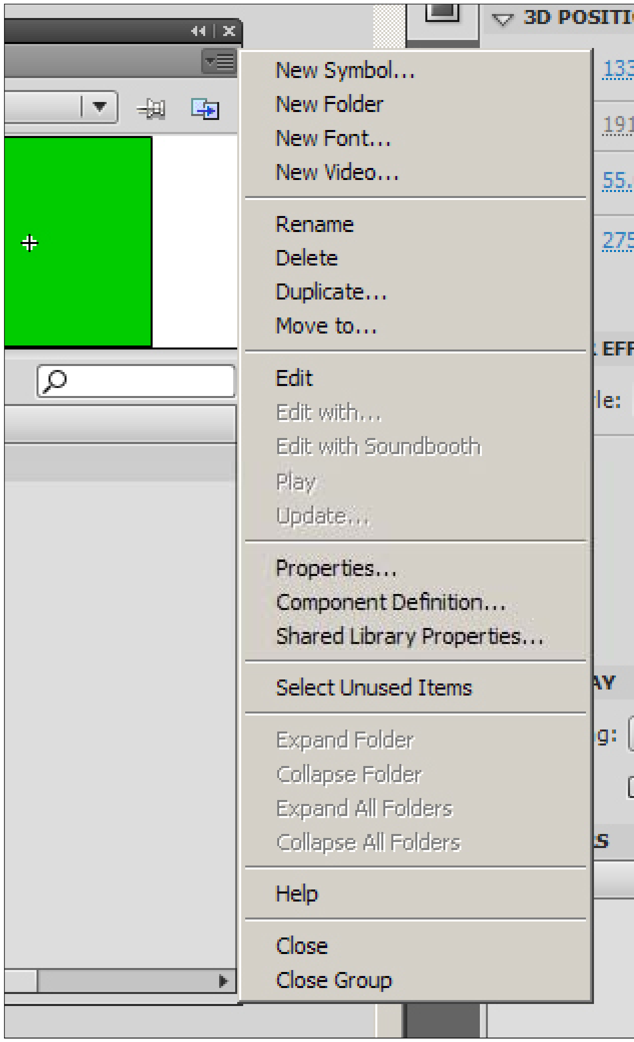
Libraries

- Libraries store symbols and assign symbols their basic behaviors.
- Every Flash file that has symbols can be a library.
- All you have to do is open a particular fla file as a library, then drag and drop symbols either onto the stage or into the current files library.
- Remember that a symbols **registration point** is dependant on where you place it on the stage.
- Also remember that the symbols in the library control the basic parameters of all instances of that symbol.



The Library Panel

- Use the buttons on the right of the panel to toggle its appearance.
- **File menu** has a number of functions that you can perform.
- Remember that <u>once you have deleted a symbol, there is no way to undo it.</u>
- Deleting a symbol <u>will cause all instances of the symbol to disappear</u>.
- Check the **usage count** before deleting a symbol.
- Can now view **multiple libraries** in a single panel.
- <u>All library items are NOT exported with .swf export.</u>



Common Libraries

- Useful for creating basic movies....
- Used to have a number of graphics and movie clips in them

Shared Runtime Libraries

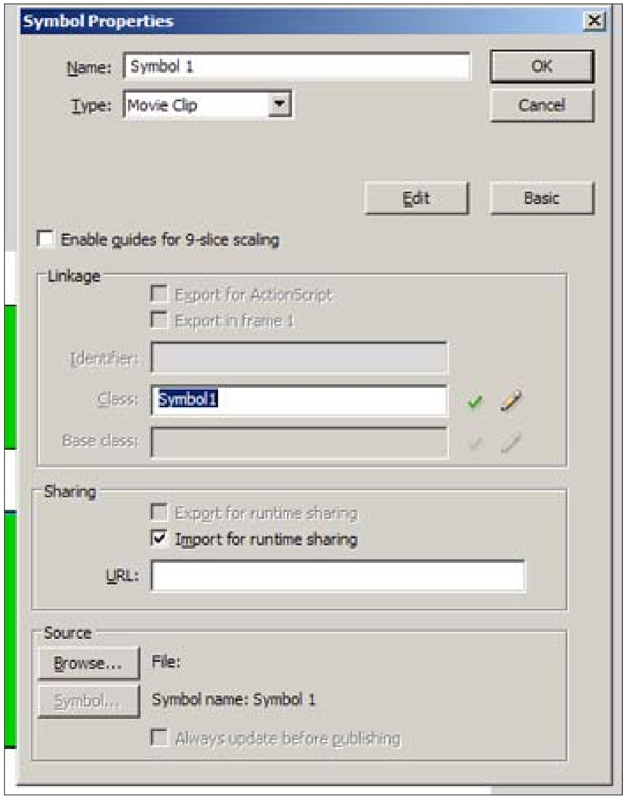
- By setting the **linkage properties** of items in the library, you can share those assets with other swfs

- Can use movie clips, fonts, etc…

- The advantage of doing this is to minimize the amount of file size in "primary" swf's

Procedure

- Create your fla that will contain the majority of your assets
- Right click each symbol in the library and choose "Linkage"
- For Linkage, select "Import For Runtime Sharing" to link to the asset in the source document.
- Enter an identifier for the symbol, bitmap, or sound that is identical to the identifier used for the symbol in the source document. Do not include spaces.



Symbol Properties

Name: Symbol 1

Type: Movie Clip

□ Enable guides for 9-slice scaling

Linkage

□ Export for ActionScript

□ Export in frame 1

Identifier:

Class: Symbol 1

Base class:

Sharing

□ Export for runtime sharing

☑ Import for runtime sharing

URL:

Source

File:

Symbol name: Symbol 1

□ Always update before publishing

Procedures (cont.)

- Enter the URL where the SWF source file containing the shared asset is posted, and click OK.
- In the destination document, do one of the following:
- Select File > Open.
- Select File > Import > Open External Library.
- Select the source document and click Open.
- Drag the shared asset from the source document Library panel into the Library panel or onto the Stage in the destination document.

Assigning Linkage Identifier to a Library Item

- Select the font item in the Library panel.
- Do one of the following:
  - Select Linkage from the Library Panel menu.
  - Right-click (Windows) or Control-click (Macintosh) the font symbol name in the Library panel, and select Linkage.
- Under Linkage, select Export for Runtime Sharing.
- In the Identifier text field, enter a string to identify the font item.
- In the URL text field, enter the URL of the SWF file that contains the font item