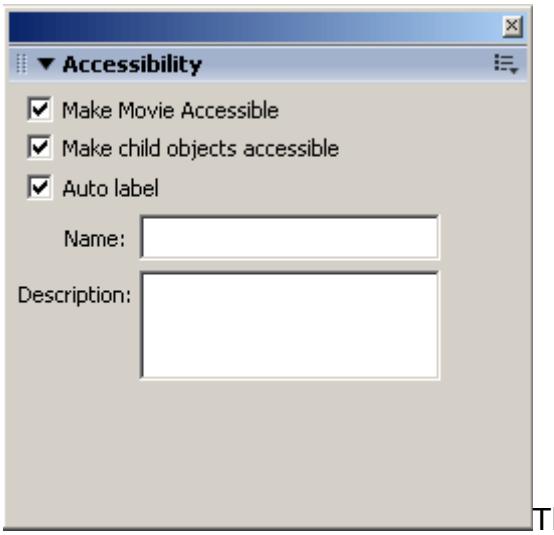
Creating Accessible Macromedia Flash Content - http://www.webaim.org/techniques/flash/keyboard.php

The Macromedia Flash accessibility panel

You can provide text equivalents for graphical elements within Flash using the Flash Accessibility panel, available at **Window > Accessibility** within Flash or by pressing **ALT** + **F2**.

Note

All of these examples use Flash MX 2004 Professional. Appearance and functionality may be slightly different in other versions of Flash.



There are several options available on the Accessibility panel. The items that are visible on the Accessibility panel vary based on the type of object that is currently selected.

Make Movie Accessible: If selected (the default), then accessibility features of Flash will be exposed to the screen reader. If not selected, the screen reader will identify that a Flash movie is there, but will not access any of the content within the movie.

Make Object Accessible: If this option is deselected, the currently selected object will be made invisible to the screen reader and the textual equivalent and any text immediately within the current symbol will not be accessible to the screen reader. This can be useful if the symbol does not convey important content. By default, objects are accessible, so be sure to select this option if the object contains important text or convey content.

Make Child Objects Accessible: If you have other objects or symbols embedded within a selected symbol, you can make them hidden by selecting this

option. This is useful for symbols that are comprised of multiple objects, but as a whole only need one text equivalent. For instance, if a movie clip symbol contains a text animation wherein each character of a word is animating independently, you wouldn't want each character to be read individually by the screen reader. To solve this, you would add alternative text to the movie clip itself and deselect the **Make Child Objects Accessible** to hide the animating items within the movie clip.

Auto label: This option tells Flash to associate buttons with text that are within or near the button. If all of your buttons have text within them or adjacent to them, then Auto label can automatically associate them and will read the text as the alternative text for the button. The results are not always as expected and care should be taken when using Auto label.

Name: This is where you would enter alternative text for a Flash object. This is what the screen reader will read in place of the selected element. You can enter a brief (one or two short sentences) name or description of the object.

Description: This is used for longer descriptions of Flash objects and is not required unless additional description is needed beyond that allowed within Name. If both Name and Description are provided, the screen reader will read the Name first, then the Description. Functionally, there is no difference between Name and Description.

Shortcut: This allows you to indicate to the user what the shortcut key is for that specific object. It does not program the keyboard shortcut, but simply alerts the user as to what the shortcut key is.

Providing Text Equivalents for the Entire Movie

If it is appropriate, you can easily provide a text equivalent for your entire movie, rather than worrying about trying to get individual parts of the movie to be accessible. This would be similar to providing an alternative text for an entire web page. Just deselect everything on the Flash stage, select the **Accessibility** panel, and deselect **Make Child Objects Accessible** to hide the internal contents of the movie. Then deselect **Auto Label** and add a brief **Name** and if needed, a longer **Description**.

Most Flash advertisements could be made more accessible in this manner, though if they contain a button, they may interfere with keyboard navigation, as described in the Keyboard accessibility section. View a movie with text equivalent ( )

Keyboard Accessibility

Unless you are using version 7 of the Flash player in Internet Explorer for Windows, when Flash receives the focus within a web page, it maintains that focus. What this means is that once you click in or tab to a Flash movie, you cannot use the keyboard to navigate to other items on the page. The screen readers that support Flash (current versions of JAWS and Window-Eyes) have built in functionality that will change focus back to the web page after all of the Flash items have been accessed. Common browsers with older versions of the Flash player, however, do not have this functionality.

View example of Flash maintaining focus

Tab and Reading Order

Because Flash is not based on linear code like HTML, the navigation and reading order of items in a Flash movie is determined roughly by their distance from a point somewhere near the upper left hand corner. In some cases, the layout of objects on the stage may cause the reading and tab order to work logically. View example of logical tab order ( ). And other times it does not. View example of illogical tab order ( ). You can test the tab order of items using your keyboard; however, you can only test the reading order of items within your Flash movie with a screen reader. If your movie has a complex layout, you can solve the problem of having to position items based upon that mystical position near the upper left of the movie by using ActionScript to specify a tab and reading order for form, button, text, and movie clip elements inside your Flash movie.

To specify a tab/reading order, two conditions must be met:

1. **All** instances of text within the movie must first be defined as input or dynamic text.
2. **All** symbol instances (occurrences of any symbol) within the movie must be given an instance name.

If you want to specify the tab order of elements within your movie, you must convert all static text objects to dynamic text object. Now add tabIndex information to a keyframe at frame 1 of your movie:

```
_root.Homepage.tabIndex = 1
_root.Contact.tabIndex = 2
_root.FirstName.tabIndex = 3
_root.LastName.tabIndex = 4
_root.SubmitButton.tabIndex = 5
```

If you are using Flash MX 2004 Professional or newer , a tab order option will display within the **Accessibility** panel. You can specify the tab order in the panel rather than using ActionScript.

In order for this to work, **all** text items must be defined as input or dynamic text and you must specify a tab order for **every** button, movie clip, and text object on the stage that has been set to be accessible in the Accessibility panel. If you miss even one, then screen readers will disregard your tabIndex altogether.

View an example of incorrect reading/tab order

View an example with tab/reading order specified with ActionScript

More examples of reading order - external link

Hiding Unimportant Flash Content

Some Flash content, including banner ads or page decorations, are often not necessary to the content of the page. Because of limitations with version 6 and older of the Flash player, including a Flash movie on a web page could, in many cases, make the rest of the HTML content on the page totally inaccessible. If the end user has an older version of the Flash player installed, the Flash movie object on the page will maintain the keyboard focus within the movie as soon as it is given focus. This means that keyboard users can become "stuck" within the Flash movie and will not be able to navigate to other elements within the page. To hide unimportant Flash content from both web browser and screen readers, add the wmode option with a value of opaque to both the object and embed tags of the web page containing the Flash movie.

```
<object ...>
<param name="wmode" value="opaque">
<embed wmode="opaque" ...>
</embed>
</object>
```

View a page with a hidden Flash movie

Detecting Screen Readers

Perhaps you want to provide a different Flash interface or options if a user is using a screen reader. For instance, you may want to provide additional buttons or turn on self-voicing features when the person watching your movie is using a screen reader. You can detect screen readers using ActionScript. The Accessibility.isActive() function will return "true" if a screen reader that is capable of accessing Flash content is detected (currently only up-to-date versions of Window-Eyes, JAWS, and IBM Home Page Reader). For instance, you might add:

```
if (Accessibility.isActive()) {
_root.selfVoicing.play();
}
```