automatically by default, the application may disable this feature if assistive technology is detected; for example:

```
import flash.accessibility.Accessibilty;
...
if (Accessibility.active)
  {
    // perform action desired when assistive technology is running,
       such as preventing the automatic playing of audio or multimedia
  myVideoPlayer.stop();
  }
```

The `flash.accessibility.Accessibility.active` property detects the presence of software querying for MSAA objects. While this generally indicates the presence of a screen reader, it is not a definitive test because other assistive technologies and software may also query for this information and give the appearance that a screen reader is running. It is also a good idea to wait a few seconds after the application loads before checking this property to ensure assistive technology has had time to query Flash Player.

Textual equivalents

When an application conveys information visually without the use of text, developers should provide a textual equivalent to assist users of screen reader software.

Image alternatives

Because screen readers cannot discern the meaning of image or animated graphic elements on the screen, developers must provide a brief text description of these elements. Text equivalents can be provided for groups of components or for individual components within an application.

Flex applications commonly present images using an Image or Loader component. To convey the contents of these components to a screen reader, use the `toolTip` attribute to specify a text equivalent. Tooltip content is also made visible to sighted users as they mouse over the image. Because `toolTip` attributes are not keyboard accessible, developers must ensure that anything conveyed in the image meets the accessibility requirements for color and contrast to ensure that all users can access the information.

```
<mx:Image width="60" height="56" source="assets/icecreampint.jpg"
    toolTip="Ice Cream Pint"/>
```

A longer, more detailed description of an image that would not be appropriate to non-screen-reader users can be provided using the `accessibilityDescription` property for the object; for example:

```
<mx:Image width="60" height="56"
  source="assets/icecreampint.jpg"
  toolTip="Ice Cream Pint
  accessibilityDescription="Our fine Ice Cream Pint provides the perfect
    serving dish for your homemade ice cream treats"/>
```

This technique can be helpful when extra detail is provided within the image itself or in a separate panel. When the description is in a separate panel, it can be difficult for a screen reader user to locate the description elsewhere on the screen, especially when it changes. Use descriptions only when needed, placing unneeded descriptions on images will make an application verbose and tedious to use.

To help screen reader users understand complex graphs and charts, an equivalent DataGrid representation or a link to an HTML page containing a data table can be invaluable. When providing equivalents for charts and graphics, be sure to provide a summary of the chart, including trends and axes, as well as detailed information if the user is likely to desire more than the summary information. Detailed information may include facts about