not using a screen reader. If the word "Error" cannot be added to the error text then an image indicating an error should be used and the accessible name of the image should be set to "Error"; for example:

```
//ActionScript
imgError.accessibilityName = "Error";
// MXML
<mx:Image id="imgError" accessibilityName="Error" />
```

When describing user interface controls that use color to convey information, use equivalent text that describes what the color signifies, not the color itself; for example:

```
//ActionScript
// A red X button with an accessible name of "Stop Process"
btnRed.accessibilityName = "Stop Process";
// a green checkmark with an accessible name of "Start Process"
btnGreen.accessibilityName = "Start Process";
// MXML
// A red X button with an accessible name of "Stop Process"
<s:Button id="btnRed" accessibilityName="Stop Process"
    skinClass="IconButtonSkin" iconUp="Red.jpg" />
// a green checkmark with an accessible name of "Start Process"
<s:Button id="btnGreen" accessibilityName="Start Process"
    skinClass="IconButtonSkin" iconUp="Green.jpg"/>
```

In the example above, note that a red X and green checkmark were used instead of more ambiguous shapes, such as a circle or square. Color alone is not sufficient to convey meaning because users with visual impairments—including color deficiencies—may be unable to distinguish the red color from the green. By using color, appropriate icons, and alternative text, the same visual information is provided in multiple ways.

Convey hierarchy through textual equivalents

The hierarchy of content is often conveyed visually, for example, by using lines in an organizational chart, multiple indents in an outline, or a pyramid shape. Developers of accessible applications must ensure that any hierarchical relationships that are conveyed visually are explained in text.

Flex allows list item tags (`li`) in HTML text, however the hierarchical nature of text in lists constructed using these items is not conveyed to MSAA or the screen reader software. Thus, hierarchical information must be conveyed via an outline structure—such as a Tree or AdvancedDataGrid component—or as a text equivalent.

Using a Tree component

In Flex, using a Tree component is the recommended approach for providing hierarchical information. The Tree component in the following example conveys a simple organization chart:

```
<mx: Text text="Contact the Appropriate Technical Support Person in Your Department" />
<mx:Tree accessibilityName="Technical Support contacts by Department" width="300"
            height="200" labelField="@label">
  <mx:dataProvider>
    <fx:XML>
      <node label="Senior Management">
        <node label="Tim Springer" data="d0"/>
      </node>
      <node label="Information Technology">
        <node label="Mary Smith" data="d1"/>
      </node>
      <node label="Finance">
        <node label="Ted Robins" data="d2"/>
       </node>
       <node label="HR">
         <node label="Recruitment">
```