```
      lblRollOverContent.text = "Content for Rollover 3";
      lblRollOverContent.visible = true;
      lblRollOverContent.tabIndex = 15;
      lblRollOverContent.setFocus();
    }
  }
}
```

The example above does not show how the rollover content is hidden, but this can be achieved using the `FOCUS_OUT` event; for example:

```
lblRollOverContent.addEventListener(FocusEvent.FOCUS_OUT, hideRollOver);


function hideRollOver(): void
{
   lblRollOver.visible=false;
}
```

Testing should be performed by users with disabilities to determine the best method of hiding rollover content. For example, if the content disappears when focus leaves the content, text would need to be added to the accessible name of the actuator explaining this.

### Context menus

Context menus, which open with a right-click (Windows) or a Control-click (Mac), change depending on the item clicked. In an accessible application, this functionality must also be available from the keyboard. Consider using a shortcut key (such as Shift+F10 on Windows) or an alternative that is accessible from the keyboard (such as a small button next to the item) to activate the context menu.

### Toolbar functionality

If a toolbar cannot be made keyboard accessible, add shortcut keys or menu items to perform the same actions. Anytime keyboard shortcuts are used, clearly document the shortcuts for the user in an accessible format. The documentation should be in the Flex application itself or in the HTML page displaying it, but it may also be provided in a user guide or other support documentation.

See "Shortcuts and mnemonics" on page 22 for more details on creating and exposing keyboard shortcuts.

### Navigation between panes

For applications with two or more panes, keyboard navigation between panes may be needed. This can be provided via a menu, shortcut keys, or the Tab key, if tabbing through all elements in one pane leads to the next pane. On Windows, for example, F6 is a commonly used shortcut key for pane navigation. For application panes with few components, tabbing through all of the components to get to the next pane is acceptable. When panes contain many items however, particularly repetitive components, developers should include a mechanism to quickly move focus past the current pane to the next pane. This allows keyboard-only users to effectively access content in any pane.

See "Implementing keyboard shortcuts" on page 22 for examples on setting keyboard shortcuts.

### Resizing, moving, and closing windows

If a window or component can be resized, moved, or closed with the mouse, a user must be able to perform those same actions using the keyboard. For example, if an object can be moved via the mouse after choosing a menu item, ensure that the menu and menu item are keyboard accessible and that the arrow keys can then be used to move the object. Alternatively, provide input fields to allow the user to manually enter new x, y, width, and height values for the object.

The following example code is from an application that displays a photograph and allows the user to crop the picture. The application provides a button, menu option, or shortcut keystroke to activate the crop feature. It