Note: While *tab order* and *reading order* are not interchangeable terms, in Flex they are both controlled through the `tabIndex` attribute of an element. Reading order is independent of a component's `tabEnabled` state. See "Reading order" on page 23 for more details.

If a tab order is not specified, a default tab order based on the horizontal and vertical coordinates of each component will be used. Depending on the application and the layout of components on the screen, this default tab order may not be logical. Explicitly setting the `tabIndex` attribute of each component will ensure a logical tab order.

Tab, Space, Enter, and arrow keys

In Flex applications, the Tab, Shift+Tab, Space, Enter, and arrow keys are the primary keys used to navigate to components and activate them. Developers must ensure their default behavior is correctly supported. The following list describes these primary keystrokes:

- Tab navigates to the next actionable component.
- Shift+Tab navigates to the previous actionable component.
- Space activates a button, checks a check box, manipulates a control, or types a space in a text field or text area.
- Enter activates a button (when the `defaultButton` property has been set) or starts a new line in a text area.
- Arrow keys (up and down) move among the choices in a menu, list component, combo box, drop-down list, or group of radio buttons (selecting them without activating them), and move among the lines in a text area.

There is no need to create event listeners to implement these key presses in Flex. The standard Flex event handler for mouse clicks is also triggered by pressing Space (and Enter if the `defaultButton` property is set). Standard Flex components also handle Tab, Shift+Tab, and arrow keys appropriately. Developers should ensure that the `tabEnabled` property is set on all actionable items that are to be included in the tab order. For standard components, this is handled automatically. For custom components, an event handler is required to process mouse clicks and relevant key presses; for example:

```
// the Mouse Click event will also monitor for the space bar to be pressed
and will be triggered by enter as well if a defaultButton has been set in the container
btnSearch.addEventListener(MouseEvent.CLICK, performSearch);
function performSearch(e:MouseEvent): void
{
  // search not shown
}
```

To allow Enter (in addition to Space and the left mouse click) to activate a button, assign the defaultButton property of a container to one button in the container. Pressing Enter while focus is on any component in the container will activate the default button. This feature works for containers such as the MX and Spark Panel as well as the MX HBox, VBox, and Form containers. It is often used for login forms to enable a user to simply press Enter after typing the username and password.

```
<s:Panel title="Default Button Example">
        <mx:Form defaultButton="{mySubmitBtn}">
            <mx:FormItem label="Username:">
                <s:TextInput id="username" width="100"/>
            </mx:FormItem>
            <mx:FormItem label="Password:">
                <s:TextInput id="password" width="100"
                    displayAsPassword="true"/>
            </mx:FormItem>
```