

provide a shortcut to the links so that screen reader and keyboard-only users can reach them without excessive tabbing; for example:

```
//Action Script
txtPageHeading.tabIndex = 1;
// additional page content not shown
txtLastPageContent.tabIndex = 90;
lnkChapter1.tabIndex= 100;
lnkChapter2.tabIndex= 101;
lnkChapter3.tabIndex= 102;
lnkChapter4.tabIndex= 103;
lnkChapter5.tabIndex= 104;

//MXML
<s:Text id="txtPageHeading" tabIndex="1" />
<!-- additional page content not shown -->
<s:Text id="txtLastPageContent" tabIndex="90" />
<mx:LinkButton id="lnkChapter1" tabIndex="100" />
<mx:LinkButton id="lnkChapter2" tabIndex="101" />
<mx:LinkButton id="lnkChapter3" tabIndex="102" />
<mx:LinkButton id="lnkChapter4" tabIndex="103" />
<mx:LinkButton id="lnkChapter5" tabIndex="104" />
```

Providing a skip mechanism

A "Skip Navigation" link, button, or shortcut provides an alternative way for users to bypass repetitive navigation links and skip to the main content. When the skip navigation link or button is activated, the focus is placed on the first component of the main content, so further tabbing and screen reader review continues from that point.

```
lnkSkipToContent.tabIndex = 1;
lnkChapter1.tabIndex = 2;
lnkChapter2.tabIndex = 3;
lnkChapter3.tabIndex = 4;
lnkChapter4.tabIndex = 5;
lnkChapter5.tabIndex = 6;
txtPageHeading.tabIndex = 10;
// additional page content not shown
txtLastPageContent.tabIndex = 100;
...
lnkSkipToContent.addEventListener(MouseEvent.CLICK, skipToContent);
function skipToContent(e:KeyboardEvent): void
{
    txtPageHeading.setFocus();
}
```

Forms

The ability of users to successfully complete forms is critical in ensuring the effective use of most Flex applications. When form label and structure information is not properly defined, users with disabilities—particularly those with visual impairments—face significant challenges in completing forms. To create accessible forms, developers must properly label fields, group fields logically, provide accessible error and instructive text, and allow sufficient time for users to complete the form.

Setting component name information - form label considerations

To be properly rendered by assistive technology, each form field used within an application must have a label associated with it. In Flex applications there are three main ways to associate labels with elements, which are described in "Setting component name information" on page 7. Developers must ensure that all form