```
//ActionScript
txtLabelReceiveEmails.text = "Would you like to receive emails?";
rbReceiveEmailsYes.accessibilityName = "Would you like to receive emails: Yes";
rbReceiveEmailsNo.accessibilityName = "Would you like to receive emails: No";
// or written with the answer first
rbReceiveEmailsYes.accessibilityName = "Yes - I would like to receive emails";
rbReceiveEmailsNo.accessibilityName = "No - Don’t send me any emails";

//MXML
<s:Text id="txtLabelReceiveEmails" text="Would you like to receive emails?" />
<s:RadioButton id="rbReceiveEmailsYes" accessibilityName =
     "Would you like to receive emails: Yes" />
<s:RadioButton id="rbReceiveEmailsNo" accessibilityName =
     "Would you like to receive emails: No" />
<!-- or written with the answer first -->
<s:RadioButton id="rbReceiveEmailsYes" accessibilityName =
     "Yes - I would like to receive emails" />
<s:RadioButton id="rbReceiveEmailsNo” accessibilityName =
     "No - Don’t send me any emails" />
```

If a RadioButtonGroup component is not present, Flex will automatically generate one as long as the `groupName` property is set on each Spark RadioButton. For this grouping to occur properly, developers must set the same value for the `groupName` property for each radio button in the group. Grouping radio buttons in this way will ensure that users can use arrow keys to cycle through all radio buttons in a group.

Note: To ensure form fields are grouped properly, developers must also set a tab order that flows through the form and any groups logically. Generally, proper tab order within a form follows the implied visual tab order or business logic of the form. The implied visual tab order is the order in which users would expect tab stops to occur, given the visual layout of the form. See “Keyboard accessibility” on page 15 and “Reading order” on page 23 for more details.

The number of radio buttons in a group is not typically indicated automatically by assistive technology for Flex applications. This is because the `groupName` property is not exposed through MSAA and assistive technologies have no way to determine if a set of radio buttons is related (other than by their relative proximity in the reading order). If the user must know the radio button count properly use the form, then it should be appended to the accessible name of the radio buttons; for example:

```
// FormItem example showing proper radio button grouping using FormItem
and RadioButtonGroup with the groupName attribute and with
accessibilityName set to override the label property to include the
ordering of the radio buttons
<mx:FormItem label="Do you want to receive emails?:">
  <s:RadioButtonGroup id="RadioGroup1" />
  <s:RadioButton label="Yes" accessibilityName="Yes (1 of 2)" groupName="RadioGroup1" />
  <s:RadioButton label="No" accessibilityName="No (2 of 2)" groupName="RadioGroup1" />
</mx:FormItem>
```

Instructive text

Instructive text that applies to the entire form should be placed at the top of the form and at the beginning of the form’s reading order. This instructive text may include any general instructions for filling out the form, as well as any instructions specific to assistive technology. Instructions such as "required fields are marked with an