Forms that are dynamically updated—including forms in which new form fields appear or become enabled based on user input—are a special case for accessibility. New form fields should appear later in the form's reading and tab order by setting the `tabIndex` as described in "Keyboard accessibility" on page 15 and "Reading order" on page 23. In general, developers should conform to the following process to ensure updates are accessible:

- Make all visible updates to the screen.
- Update all accessibility information via ActionScript.
- Make a single call to `flash.accessibility.Accessibility.updateProperties()` to notify any active assistive technologies that the screen data has updated.

Note: Do not call the `Accessibility.updateProperties()` method more than once per second. When this function is called assistive technologies update their cached version of the accessibility data, which can be processor intensive. Frequent updates can cause the system to slow down. The best practice is to make all required changes to accessibility data and then make a single call to the `Accessibility.updateProperties()` method. As a reminder, if the convenience accessor methods are used, there is no need to call `updateProperties()`, as it is done automatically. See "Reading and tab order in practice" on page 29 for more details.

Focus

A component is said to have *focus* when it is the center of user interaction. Focus includes three separate but highly interrelated concepts: visual focus, keyboard focus, and programmatic focus. Visual focus refers to a visual cue—such as a yellow or blue rectangle or cursor—that indicates where the next user interaction will take place. Keyboard focus refers to the location at which an action will be performed based on the next user interaction from the keyboard. Programmatic focus is exposed to assistive technology to indicate where user interaction will take place.

When focus is properly indicated and maintained, the user knows where they are in a program and what their next keystroke will do. When an application does not maintain focus properly, most users will be confused and frustrated, and assistive technology users will likely be unable to use the application at all.

In almost all situations, visual, keyboard, and programmatic focus are set to the same component. The FocusManager class manages the focus for Spark and MX components in response to keyboard and mouse activity. This can be seen when focus is moved to the appropriate element when a user presses Tab or Shift+Tab. In general, as long as developers provide keyboard accessibility and set the tab order for applications properly—procedures documented in "Keyboard accessibility" on page 15 and "Reading order" on page 23—Flex applications will provide proper focus control.

While this automatic control is sufficient for most situations, sometimes developers may need to set focus to an element explicitly. For example, when a form is submitted and an error is detected, focus should be moved directly to the error message provided at the top of the form. Similarly, when a new window appears, focus should be placed on the first interactive form field. In these situations developers need to programmatically set focus. Unexpected focus changes, however, should generally be avoided, as they can confuse and frustrate users who cannot visually perceive the change in focus or who must use the keyboard to return to the active component prior to the focus shift.

Avoid forced focus changes

When focus is changed programmatically from one element to another based on certain conditions being fulfilled, it is said to be *forced*. For example, consider a phone number entry form that contains three form fields for entering the area code, exchange, and extension. After the user types the third number into the area code field, the application moves the focus automatically to the next field. Because this type of unexpected focus change can be disorienting for users of assistive technology who cannot see the visual indications of the change, forced focus changes are to be avoided.