

FormHeading control are placed in the reading order along with the form fields. This can cause field labels to be read twice by screen readers because both the FormItem label and the component inside the FormItem have the same accessible name and, by default, both are read. To address this, silence the FormItem container's label by setting the `accessibilityEnabled` property to `false` or the `silent` property to `true`; for example:

```
// where formItem1 is the form item that should be made silent to assistive technology
// check to see if accessibility properties is present for the FormItem's label
    and create it if not
if (formItem1 && formItem1.itemLabel && !formItem1.itemLabel.accessibilityProperties)
    formItem1.itemLabel.accessibilityProperties = new AccessibilityProperties();
//Set the label to be silent - indicating it should not be read in a screen reader
if (formItem1 && formItem1.itemLabel)
    formItem1.itemLabel.accessibilityProperties.silent = true;
```

When a form contains many FormItem containers, developers can populate an array with each FormItem container, loop through the array, and set `silent` property to `true` for each item in the array; for example:

```
// array of all form items to be used to set the accImpl to silent
var col:ArrayCollection = new ArrayCollection
    ([f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12,f13,f14,f15,f16,f17]);
var i:int = 0;
// loop through all form items and set the form item label to silent
i = 0;
while (col && i < col.length)
{
    if (col[i] && col[i].itemLabel && !col[i].itemLabel.accessibilityProperties)
        col[i].itemLabel.accessibilityProperties = new AccessibilityProperties();
    if (col[i] && col[i].itemLabel)
        col[i].itemLabel.accessibilityProperties.silent = true;
    i++;
}
```

The FormHeading control is automatically added to the accessible name of each item in the form. Thus, as with the FormItem container's label component, the FormHeading should also be made silent. The FormItem component itself could not be made silent, because it is a container for the form fields and disabling accessibility on it would disable accessibility on its child components. In contrast, the FormHeading control has a direct accessibility representation, so developers can quickly set it to silent without having to access any child components; for example:

```
// set the accImpl on the form heading (fh) to silent
if (!fh.accessibilityProperties)
    fh.accessibilityProperties = new AccessibilityProperties();
fh.accessibilityProperties.forceSimple = true;
fh.accessibilityProperties.silent = true;
```

## Notes on hiding content

It is important to remember that content hidden by setting `accessibilityEnabled` to `false` is only hidden from certain classes of assistive technologies—most commonly screen readers. This technique does not hide the content from sighted users who may have other disabilities. Thus, content hidden in this way must still comply with other accessibility guidelines.

## Reading and tab order in practice

Both tab order and reading order are controlled by setting the `tabIndex` property. All components that have `tabEnabled` or `focusEnabled` set to `true` will receive keyboard focus in the order defined by the `tabIndex` property. Likewise, all components that have `enableAccessibility` set to `true` (or `silent` set to `false`) will be read by screen reader software in that same order.