

SAP Hows About Screen Reader Support in SAP Fiori

July 5, 2018

When we talk about software design, it is often about the visual experience of an application. Yet there are users who experience design without seeing, but by using their auditory and tactile system. Screen readers “read out” the content of the user interface by speech output or braille display. SAP UI5 offers screen reader support to aid people with visual impairments. The implementation is based on the [WAI-ARIA](#) (Accessible Rich Internet Applications) standard. The majority of [UI5 controls](#) provide screen reader support on desktop platforms (Windows, Mac OS).

Screen reader navigation and interaction support in the SAP Fiori launchpad and in SAP Fiori apps is based on four principles: exploring, operating, notifications, and prerequisites for accessibility support. Let’s look at how SAP Fiori works with a screen reader before giving recommendations on how to ensure that your app is screen reader ready.

Exploring

When accessing an SAP Fiori app for the first time with a screen reader, you’ll start with exploring to get an overview. Like a typical explorer, the screen reader looks around what’s on the UI: sections, forms, toolbars, headings, tables.

Some screen readers offer virtual mode support, such as the ‘Virtual PC Cursor Mode’ in Jaws or ‘Browse Mode’ in NVDA. This is especially useful for pages that contain much non-focusable content. It can also be used in certain complex controls for orientation within. Visually hidden controls, tables used for layout purposes and decorative content like images wouldn’t be announced.

Not all screen readers offer virtual mode support, but offer other techniques like extra keys for structure and extended focus modes, such as in Windows Narrator.

If a screen reader has respective functions like overview dialogs, list features or designated navigation keys, controls with similar behaviors can be accessed in groups, special dialogs or by designated screen reader keys in virtual modes. To enable this, SAPUI5 controls are attributed with the correct ARIA role. Jaws, for example, allows access to a list of headings. Then you can directly navigate to the next or previous heading or use an overview dialog.

In the same way, you can orientate on and navigate by regions. The number of regions and their titles are app-specific. A region might be a banner area, a header region, a master region, a main region, a details region or a complementary region, which may show up empty and become a more precise context later.

Operating

Now that you know where everything is on the UI, you can now start interacting. Want to select a list item? You know where to navigate to.

For this, use the keyboard focus mode or a mode the screen reader may switch to when reaching interactive content. Some screen readers support various modes, for example, Jaws offers the ‘PC Cursor Mode’ and ‘Application Mode’.

Navigation to and interaction with active controls follows implemented keyboard support. For successful operation of screen readers, keyboard Navigation and interaction support has to be present and implemented for all active UI controls. The tab key that triggers navigation between different controls is very important. Using the arrow keys, you can navigate within controls.

Each focused control's label, role (control type), property (state, value, text, tooltip, etc.), and additional info like usage help or tool tips should be spoken. Labels briefly describe the control. The role informs whether you are dealing with an editable field, a button, a link, etc. States signalize whether it is possible to edit the field, a button is pressed, or a link is disabled.

Notifications

Launchpad notifications, application messages, and dialogs, state transitions and updates, for example loading information, need to be passed on to the screen reader immediately. When regions or the entire app are busy, ARIA attributes like `aria-live` or `aria-busy` would be used. Also, important control updates as a result of control operation are announced, like when you've marked a check box or changed a value in a combo box.

Prerequisites

Screen readers get the information directly from the DOM (Document Object Model) and the platform accessibility API, which is used by the native application or the browser. In general, no special SAP Fiori launchpad settings or special screen reader configuration should be required for usage. However, when creating new controls or changing the HTML structure of existing ones, check the validity of the resulting HTML. This also counts for information about its role, label, values etc.

The first thing that is read upon entry for any control is its ARIA role. Proper labeling of all UI elements is a prerequisite so that the screen reader can announce everything correctly. All edit boxes, search fields and column headers need labels, for example:

- `labelFor="..."`
- `aria-label="..."`
- `aria-labelledby="..."`
- `placeholder="..."`
- `title="..."`

Visual labels on the UI need to be semantically associated with the control they describe. If it is done correctly can be easily checked: If you click on a label, the focus will move on its UI element.

Complex controls like pages, panels, tables, and input controls need a title. As you can see in the example below, you can use the standalone title control to label a radio button, but it needs to be associated with it.

```
<mvc:View
  xmlns:mvc="sap.ui.core.mvc"
  xmlns="sap.m">
<Title id="rbGroupTitle" text="Select an option" textAlign="Center"/>

<RadioButtonGroup columns="2" selectedIndex="2" ariaLabelledBy="rbGroupTitle">
  <buttons>
    <RadioButton id="RB2-1" text="Option 1"/>
    <RadioButton id="RB2-2" text="Option 2" editable="false"/>
    <RadioButton id="RB2-3" text="Option 3"/>
  </buttons>
</RadioButtonGroup>
</mvc:View>
```

For headings (table toolbar, page header, form toolbar, panel toolbar), the text should be in a `sap.m.Title` control. Here you can see an example for a dropdown list:

```

<mvc:View
  xmlns:mvc="sap.ui.core.mvc"
  xmlns="sap.m">
<Panel expandable="true">
  <headerToolbar>
    <Toolbar height="3rem">
      <Title text="Header"/>

      <ToolbarSpacer/>
      <Button icon="sap-icon://settings"/>
      <Button icon="sap-icon://drop-down-list"/>
    </Toolbar>
  </headerToolbar>
  <content>
    <Text text="Lorem ipsum"/>
  </content>
</Panel>
</mvc:View>

```

In both examples, you can see that the labels ‘Header’ and ‘Select an option’ begin with a capitalized word and end without a period since it is no sentence. This helps your screen reader to read the label with the appropriate inflection.

The label should describe the control as brief as possible. Often, a single word such as ‘Add’, ‘Remove’ or ‘Search’ is sufficient. In the first example, the label ‘Select an option’ is a short phrase so that the control can be identified properly.

If the app uses technologies or components such as diagrams and interactive graphics, then screen reader accessible alternatives need to offer the same content and functionality. Also, tooltips and semantic colors require special ARIA labeling so that the screen reader can interpret them correctly.

Read up more on control specific behavior in the [documentation](#).

All accessibility features described in this article refer to SAP Fiori Design realized with SAP UI5’s technical prerequisites for accessibility support. This article does not give a statement on the accessibility support of any SAP product. The realized implementation within an individual SAP Fiori application or an SAP UI5 control might differ. Please contact accessibility@sap.com to learn about the accessibility support of a certain SAP application by requesting an Accessibility Status Document.