MAF Configuration

Conditional UI behavior

August 2022

This document covers multiple ways to conditionally control the UI display of a component. In this example the component being conditionally controlled is a <smart-input>.

A <smart-input> component has these 3 boolean properties that affect its display within the UI, readonly, hidden and required. In a prior configuration example, we added the Risk field to the WO Create page of Technician (TECHMOBILE). This Risk field could be conditionally controlled based on the value of the WO Priority value that is also on this same page.

To do these examples start by changing the Create WO page:

You add Risk to the dsCreateWo schema:

<attribute name="risk"/>

You add Risk (after display of Priority) to the page using a smart-input within a Box:

<box children-sizes="100" direction="row" fill-child="true" fill-parent="true"

 padding-bottom=".5" padding-top=".5">

 <smart-input hide-step-buttons="true" label="Risk" value="{dsCreateWo.item.risk}"

 />

</box>

There are multiple ways to conditionally set the readonly, hidden and required property for the risk field. Assuming a Priority value can be between 1 and 5 where a value of 1 or 2 is considered high, the following conditions could be done for WOs with a priority that is High (priority < 3).

1. You can use an inline condition to set the values of each of these properties dynamically.

you could make the Risk field required when Priority is high:

<smart-input hide-step-buttons="true" required="{dsCreateWo.item.wopriority < 3}" label="Risk" value="{dsCreateWo.item.risk}"/>

you could make the Risk field hidden when Priority is high:

<smart-input hide-step-buttons="true" hidden="{dsCreateWo.item.wopriority < 3}" label="Risk" value="{dsCreateWo.item.risk}"/>

you could make the Risk field read only when Priority is high:

<smart-input hide-step-buttons="true" readonly="{dsCreateWo.item.wopriority < 3}" label="Risk" value="{dsCreateWo.item.risk}"/>

1. You can use a state variable to drive the value of each of these properties dynamically. Using a state variable enables this condition to be applied to many fields (not just risk) using the same variable.

Add a Page State Variable for use with the Risk field. This example uses 1 state variable for all 3 properties (required, readonly & hidden), you could choose to have separate variables for each.

Define a Page State variable

 <state name="riskFlag" type="boolean" value="{false}"/>

On the Priority smart-input field you would add:

 on-change="{page.state.riskFlag=event.target.value<3}"

this will set the state variable, riskFlag, to true when the priority field is changed to a value less than 3 (1 or 2).

you make the Risk field required when Priority is high using the state variable by adding this to the smart-input field for risk:

 required="{page.state.riskFlag}"

you make the Risk field hidden when Priority is high using the state variable by adding this to the smart-input field for risk:

 hidden="{page.state.riskFlag}"

you make the Risk field read only when Priority is high using the state variable by adding this to the smart-input field for risk:

 readonly="{page.state.riskFlag}"

1. You can use a calculated datasource field to drive the value of each of these properties dynamically. One additional benefit of using a datasource field is that this field value is available to javascript code via the datasource, if needed.

A calculated attribute added to the dsCreateWo data source and set to True when the WO Priority value is less than 3. This example uses 1 schema field for all 3 properties (required, readonly & hidden), you could choose to have separate field for each. The riskFlag will be True when the Priority is high (1 or 2).

 <attribute name="riskFlag" type="BOOL" sub-type="YORN" >

 <![CDATA[return item.wopriority < 3;]]>

 </attribute>

you make the Risk field required when Priority is high using the calculated datasource field:

 <smart-input hide-step-buttons="true" required="{item.riskFlag}" label="Risk" value="{dsCreateWo.item.risk}"

 />

you could make the Risk field hidden when Priority is high using the state variable:

 <smart-input hide-step-buttons="true" hidden="{item.riskFlag}" label="Risk" value="{dsCreateWo.item.risk}"

 />

you could make the Risk field read-only when Priority is high using the state variable:

 <smart-input hide-step-buttons="true" readonly="{item.riskFlag}" label="Risk" value="{dsCreateWo.item.risk}"

 />