

## Custom Property Examples

Incidents can have different forms (each displaying a different set of attributes) based on incident type and workflow stage.

```
incidentFormSelection = (type, division, stage); (type, division); (type, stage); (type); (default)
```

This property's value is a semi-colon-delimited list of items. Each item describes components of a form name.

### **(type, division, stage)**

First, the system tries to find a form with Incident\_typeValue\_divisionValue\_stageValue. If found, it displays the "Incident Details" tab (with id "MoreInformation") contents from the selected component.

If stageValue is null (incident not submitted), it will ignore the triplet and falls back to the next item (type, division). However, if (type, division, null) is found in the triplet itself, the system will try to pick a component with Incident\_typeValue\_divisionValue.

If nothing matches, the system falls back to the next item (type, division) and then to (type) and, finally, the system uses the default form to display the contents.

### **(type, division)**

The system tries to find the component with Incident\_typeValue\_divisionValue. If divisionValue is null, it tries to find a component with "Incident\_typeValue". Because type is mandatory, typeValue cannot be null.

### **(stage, division)**

First, the system tries to find a component with "Incident\_stageValue\_divisionValue". If stageValue is null, try to find Incident\_divisionValue, else if divisionValue is null, the system tries to find Incident\_stageValue. If nothing is found, it falls back to the next item in the token (type, stage).

```
incidentReadOnlyFormSelect = (type, division, "ReadOnly"); (type, "ReadOnly"); (default, "readonly")
```

### **(type, division,"ReadOnly")**

First, the system tries to find a component with "Incident\_typeValue\_divisionValue" so that it can reuse the existing form in read-only mode. If not found, it tries to find Incident\_typeValue\_divisionValue\_ReadOnly component. If this is not found, the system falls back to the next item in the mapping (type,"ReadOnly").

If Incident\_typeValue is found, the system will use it. Otherwise, it tries to find Incident\_typeValue\_ReadOnly. At last, if no match is found, it will fall back to the default.

Note that divisionValue could be null. If it is, the system will try to find Incident\_typeValue component. If not found, it tries to find Incident\_typeValue\_ReadOnly. If not found, it falls back to the next item in the list.