FORMS

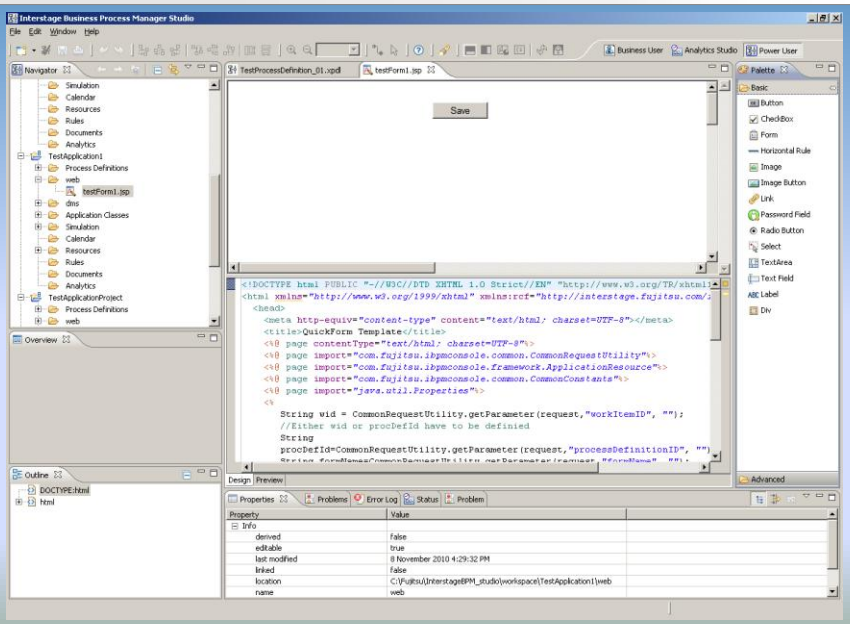
Forms

- Forms can be assigned to
  - Start node
  - Activity node
  - Voting node
- One Form can be reused at many Nodes
- Any node may have one or more forms
  - If multiple forms are added, they are displayed as tabs
- Default Form
  - Default form is displayed if no form is assigned

Form Editor

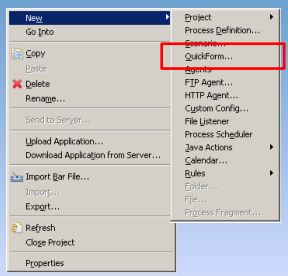
- Drag-n-drop to create rich interactive forms easily
- Maintain forms as part of BPM project, package and deploy
- Pre built widget to access and manipulate UDAs
  - Basic and Advance widgets
  - Layouts
  - widget properties, style
  - Define events and JavaScript functions
- Creates JSP forms
  - JSPs can be generated and extended to add more functionality
- Edit Source in source code view
- Preview forms in studio

Interstage BPM Forms Designer

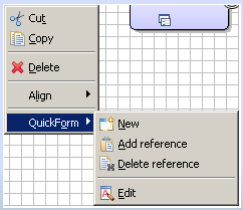


Selecting Form Design Menu

- Create form
  - New -> Quick Form

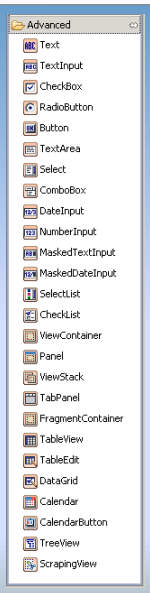


- Create and attach form to Node
  - Right Click on **Node**→ QuickForm→New



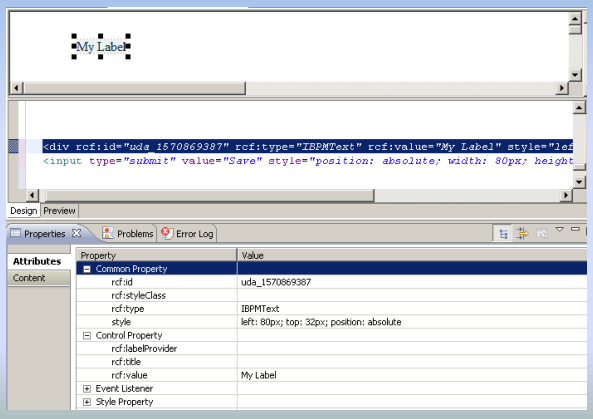
Control Palette

- There are two sets of widget
- Basic
  - Basic HTML widget
- Advanced
  - HTML widget with associated actions to access UDAs
  - In-built validation support



Add Widgets

- Drag and drop controls on the panel
- Configure widget Properties
  - Property Attributes
  - Edit Source



Common Widget

- Text
- Text Input
- Date Input
- Checkbox
- Select
- Table

Widget : Text

FUJITSU

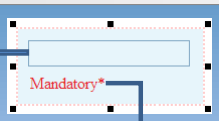
- Used to display read-only text or labels
- Text can be static or a UDA value

```
<div rcf:id="uda_2019988988" rcf:type="IBPMText" rcf:xpath="" rcf:value="Text"
                        style="left: 112px; top: 125px; position: absolute">
</div>
```

Tag	value
rcf:id	UDA name with prefix “uda” e.g. “*uda_CustomerName*”
rcf:type	control type identifier
rcf:xpath	XPath expression, if UDA is XML type
rcf:value	default value of label text. If UDA is assigned, UDA value will replace default value

Widget : TextInput

FUJITSU



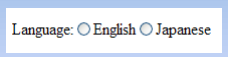

```
<div style="position: absolute; width: 188px; height: 78px; left: 237px; top: 82px“>
    <div rcf:id="uda_1563017640" rcf:type="IBPMTextInput" rcf:xpath="" rcf:width="155px“
            rcf:height="20px" style="left: 12px; top: 12px; position: absolute“
            rcf:mandatory="true" ></div>
    <div rcf:id="error_1563017640" rcf:type="IBPMText" rcf:value=“ Mandatory*"
            rcf:color="#FF0000“ rcf:width="155px" rcf:height="20px" style="left: 14px; top:
            46px; position: absolute"></div >
</div>
```

Tag	value
rcf:id	UDA name with prefix “uda” e.g. “*uda_CustomerName*” and with prefix “error” for validation control e.g. “ *error_CustomerName*”
rcf:mandatory	True/false

Widget : CheckBox & RadioButton

FUJITSU

- Radio Button and Check box group
  - It's not allowed to create multiple widgets with same id, that puts a constraint on creating Widget Group
    - **Constraint**: form cannot have two widget with same ***UDA***, rcf IDs are unique.
    - **Solution**: use basic HTML control with UDA values



- Creating group in BPM Form

```
<span rcf:type="IBPMText" rcf:value="Language:" ></span>
<input type="radio" NAME="uda_Variable1" value="English"
        <%=(ibpmUDAprop.getProperty("uda_Variable1").equals("English"))?"checked":"" %>>
        <span rcf:type="IBPMText" rcf:value="English" ></span>
<input type="radio" NAME="uda_Variable1" value="Japanese"
        <%=(ibpmUDAprop.getProperty("uda_Variable1").equals("Japanese"))?"checked":"" %>>
<span rcf:type="IBPMText" rcf:value="Japanese" ></span>
```


Widget : Select

FUJITSU

Tag	value
rcf:id	UDA name with prefix "uda" e.g. "*uda_Country*" and with prefix "error" for validation control e.g. " *error_Country*"
rcf:options	semicolon separated list of values e.g. "USA;UK;AUS"
rcf:prop_options	Name of UDA with list of values

Widget : Date Input

FUJITSU

- Date widget can be combed with Calendar widget for easy input
  - Display and update Date UDA
  - Date Popup Button displays Calendar


```
<div rcf:id="uda_ReqDate" rcf:type="IBPMDateInput" rcf:value="Requestor Date:"
                                    rcf:target="IBPMPopupCalendar_0885822"></div>
<div rcf:id="IBPMPopupCalendar_0885822" rcf:type="IBPMPopupCalendar"
                                    rcf:targetDateInputId=" uda_ReqDate"></div>
<div rcf:id="CalendarButton_0885822762" rcf:type="CalendarButton"
                                    rcf:target="IBPMPopupCalendar_0885822" ></div>
```


Widget : Table View and Edit

FUJITSU

- Used for displaying XML (UDA) in tabular format.
  - Table View is read only
  - Table edit allows editing values
    - Editing cell values results in updating XML element values
    - Adding/deleting rows in table results in updating XML (UDA) structure.

Table View

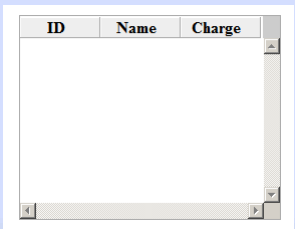


Table Edit

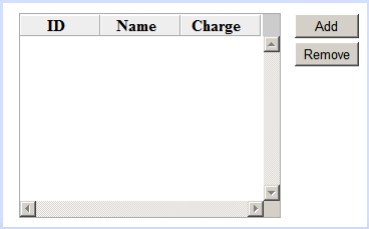


Table Edit

FUJITSU

```
<div style="position: absolute; width: 495px; height: 354px; left: 52px; top: 641px">
    <div rcf:id="uda_XXX" rcf:type="IBPMTableEdit" rcf:xpath="" rcf:width="400px" rcf:height="300px" >
        <div rcf:id="IBPMViewColumn_0878587072" rcf:type="IBPMViewColumn" rcf:name="col1" ></div>
        <div rcf:type="IBPMViewColumn" rcf:name="col2"></div>
    </div>
    <div rcf:type="IBPMButtonForTable" rcf:label="Add" rcf:targetTableId="uda_XXXX" ></div>
    <div rcf:type="IBPMButtonForTable" rcf:label="Remove" rcf:targetTableId="uda_XXXX" ></div>
    <div rcf:id="error_0878587072" rcf:type="IBPMText" rcf:value="*" rcf:color="#FF0000" ></div>
</div>
```

Tag	value
rcf:id	uda name with prefix “uda” e.g. “*uda_dataXML*” and with prefix “error” for validation control e.g. “ *error_dataXML*”
rcf:name	Name of IBPMViewColumn control should match the XML element name
rcf:targetTableId	Target for button action, UDA name or the id of table control.

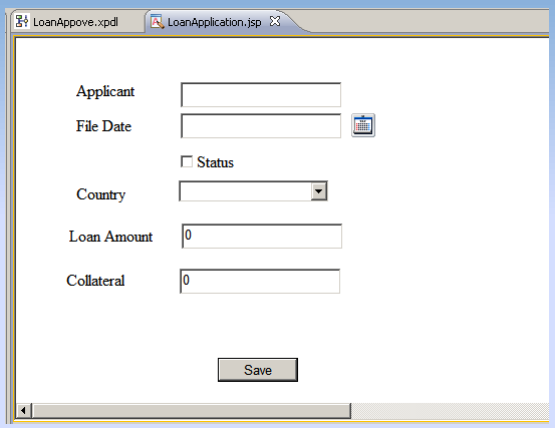
UI Container

FUJITSU

- View Container
- Panel
  - Container with title and body
- View Stack
  - Used to switch display in the same section of screen
- Tab Panel
- Fragment Container
  - Specify container information in a separate file

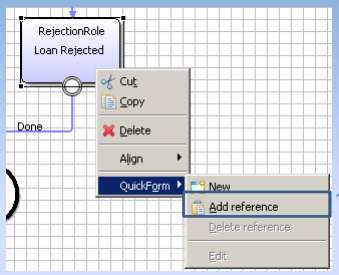
Sample Form

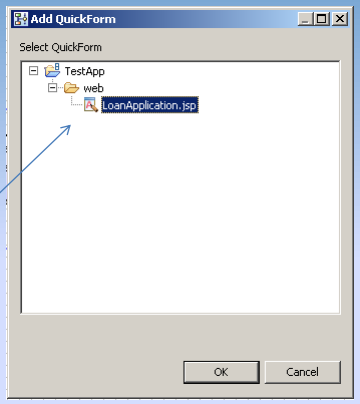
- Example Loan Application Quick Form



Assign Form reference to Node

- Assign LoanApprove QuickForm to all Nodes
  - Add reference

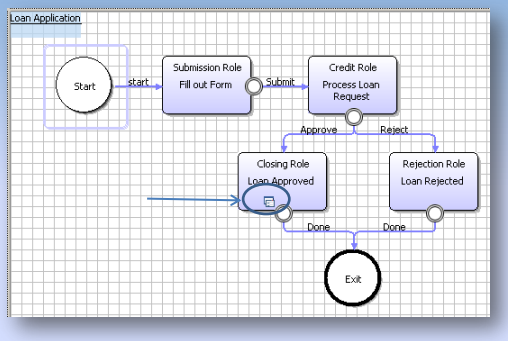




Add Form Reference to Node(s)

FUJITSU

- Form Icon
  - Form Icon will appear on nodes with Forms assigned



Widget Properties

FUJITSU

- Common Property
  - id, style, type
- Control Property
  - enabled, read-only, password, max length
- Event Listener
  - Mouse of key event listeners e.g. onBlur, onFocus, onChange, onKeyPress
- Style Property
  - Look and feel, color, padding, border etc.
- Validation Property
  - Mandatory: true/false

Event Listener

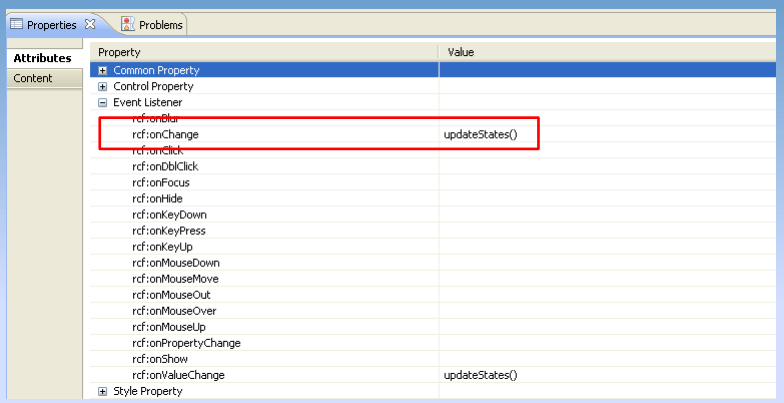
FUJITSU

- HTML/JavaScript event listener and action framework is a powerful functionality to provide interactive rich UI experience to users.
- Capture events based on user actions and perform actions to manipulate the UI as required.

- example:
  - On selecting Country from one Select/Combo-box, change list of states in another select box.
  - Event used: onChange/onValueChange
  - call JavaScript function “updateStates()” to update the values in 2nd select box.

Event Listener

FUJITSU

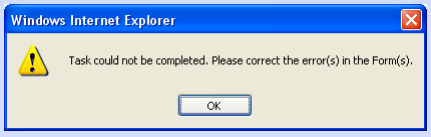



```
<div rcf:id="uda_1872941250" rcf:type="IBPMTextInput" rcf:xpath="" rcf:width="155px"
rcf:height="20px" style="left: 12px; top: 12px; position: absolute" rcf:onChange="updateStates()“
></div>
```

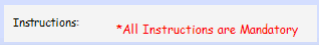

Form Validation

FUJITSU

- Form is validated for errors when
  - Form is saved
  - User makes choice to complete the task
- All controls with “error” tag are evaluated and if validation fails, JavaScript message is displayed:
- JavaScript error message:



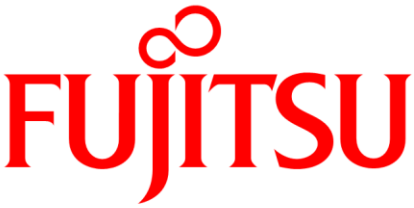
- Form Error Message:



Form Submission and Choices

FUJITSU

- Choice buttons are generated using outgoing arrows from the node
- Choices are displayed in *Choice* panel below form panel
- “Save” button is generated by default on the form.
- On Save, any change on the form is saved and UDAs are updated.
- On Submit (make choice), UDAs are updated and task is completed.



shaping tomorrow with you