

# Basic guide for using IBM Web Experience Factory

Priyoko T. Sudarmadi ([priyoko.sudarmadi@anabatic.com](mailto:priyoko.sudarmadi@anabatic.com)), PT. Anabatic Technologies

Vitri ([vitri@anabatic.com](mailto:vitri@anabatic.com)), PT. Anabatic Technologies

June 2012

© Copyright International Business Machines Corporation 2012. All rights reserved.

**Abstract:** Learn how to use IBM Web Experience Factory 8; specifically, how to create a simple Create, Retrieve, Update, Delete (CRUD) portlet and how to add a filter to the portlet, using one or two input text or combo boxes. This white paper also explains the steps to add validation rules to the portlet and to add notifications to the portlet input form. We focus on the step-by-step process and provide images for all steps.

## Table of Contents

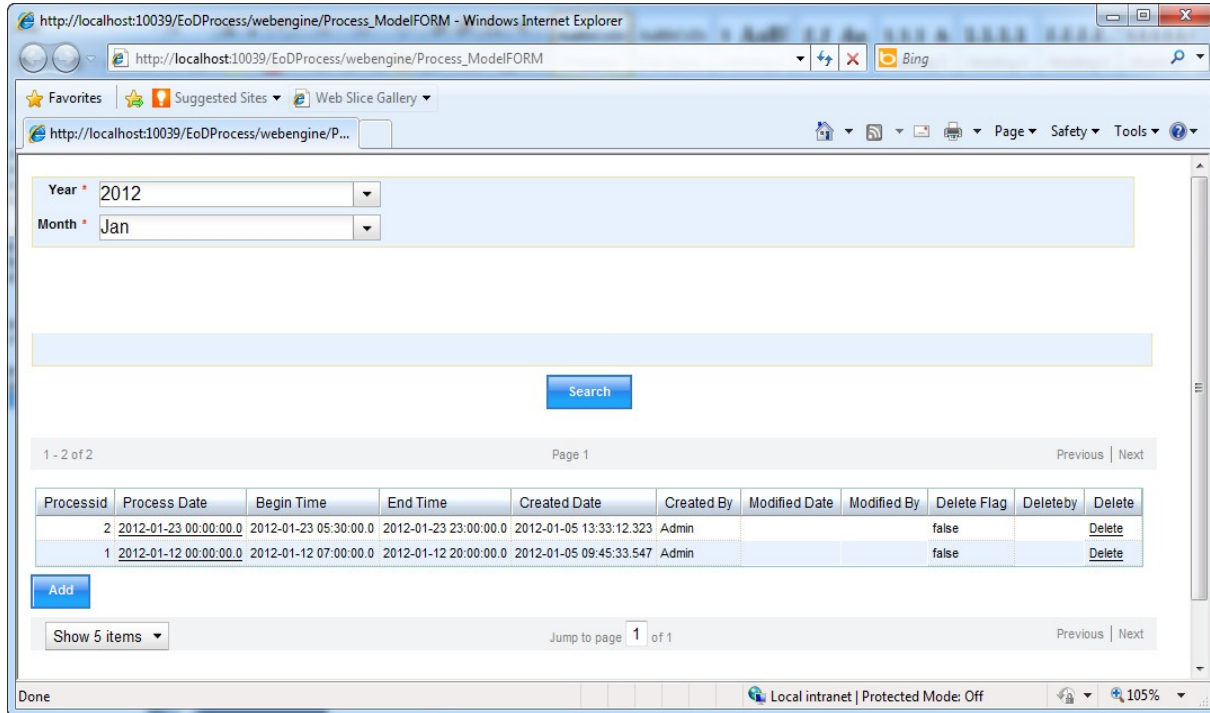
<a href="#">1 Creating a simple CRUD portlet.....</a>	<a href="#">2</a>
<a href="#">1.1 Create a WPF Project.....</a>	<a href="#">2</a>
<a href="#">1.2 Create a “Database Service Provider” Portlet Factory Model.....</a>	<a href="#">8</a>
<a href="#">1.3 Create a “List and Detail Service Consumer” WPF Model.....</a>	<a href="#">22</a>
<a href="#">1.4 Create Form.....</a>	<a href="#">27</a>
<a href="#">1.5 Update.....</a>	<a href="#">33</a>
<a href="#">1.6 Delete.....</a>	<a href="#">35</a>
<a href="#">2 Filter using select input.....</a>	<a href="#">38</a>
<a href="#">2.1 Edit the Data Field Settings Builder.....</a>	<a href="#">38</a>
<a href="#">2.2 Edit View and Form ProcessFORM_ServiceView.....</a>	<a href="#">41</a>
<a href="#">3 Add field validation in the portlet input form.....</a>	<a href="#">45</a>
<a href="#">3.1 Add the Rich Data Definition Builder for input form.....</a>	<a href="#">47</a>
<a href="#">3.2 Add the Rich Data Definition Builder for list form.....</a>	<a href="#">50</a>
<a href="#">3.3 Add the Rich Data Definition Builder for edit form.....</a>	<a href="#">55</a>
<a href="#">3.4 Enable input validation in input form.....</a>	<a href="#">60</a>
<a href="#">4 Add notifications in the portlet input form.....</a>	<a href="#">63</a>
<a href="#">4.1 Add a delete notification.....</a>	<a href="#">63</a>
<a href="#">4.2 Sorting.....</a>	<a href="#">66</a>
<a href="#">4.3 Enable/disable paging.....</a>	<a href="#">67</a>
<a href="#">5 Conclusion.....</a>	<a href="#">69</a>
<a href="#">6 Download.....</a>	<a href="#">69</a>
<a href="#">7 Resources.....</a>	<a href="#">69</a>
<a href="#">8 Author biographies.....</a>	<a href="#">70</a>

# 1 Creating a simple CRUD portlet

In this section we explain the steps for creating a simple Create, Retrieve, Update, Delete (CRUD) portlet in IBM® Web Experience Factory (WEF) 8, formerly IBM WebSphere® Portlet Factory (WPF).

**NOTE:** Although we use WPF v7 for the screenshots, we tested the steps with WPF v6 and the latest WEF v8, and they complied. For deployment, you can use IBM WebSphere Portal Server v6.1 or WebSphere Portal Server v7.

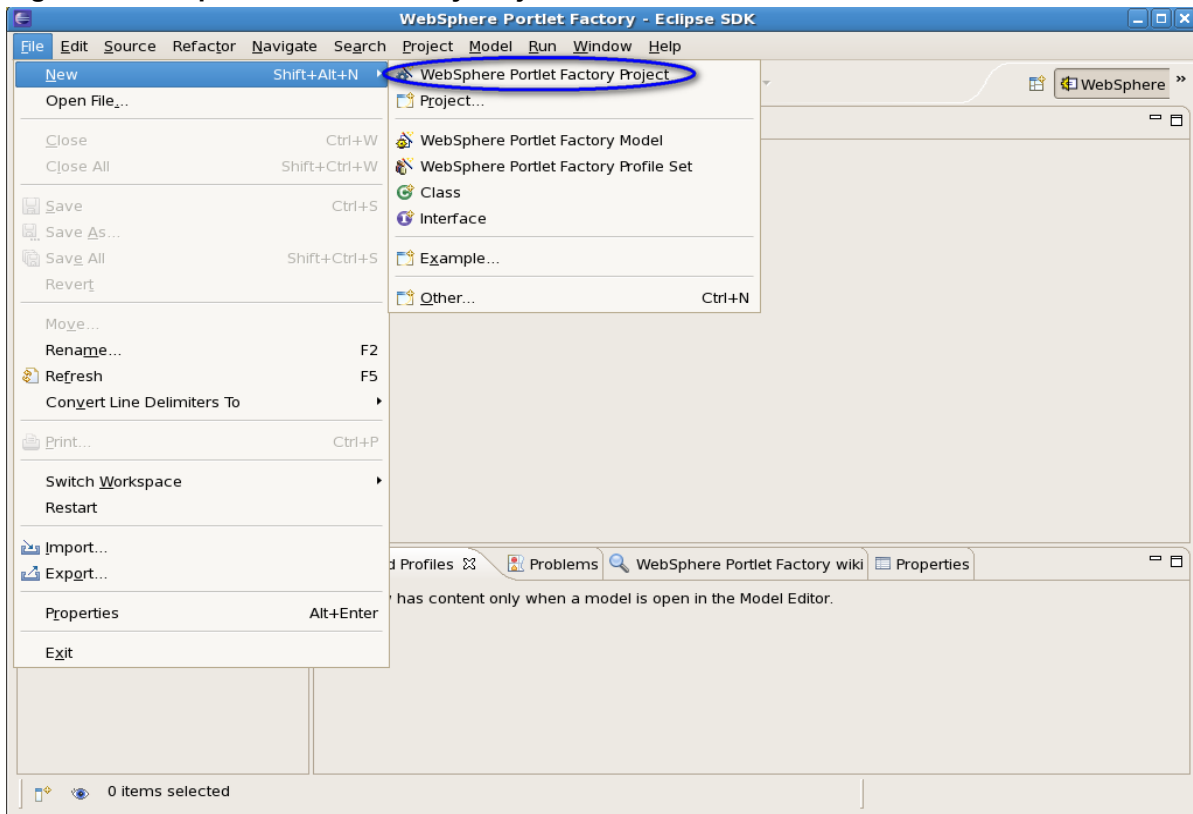
**Figure 1. Simple CRUD portlet**



## 1.1 Create a WPF Project

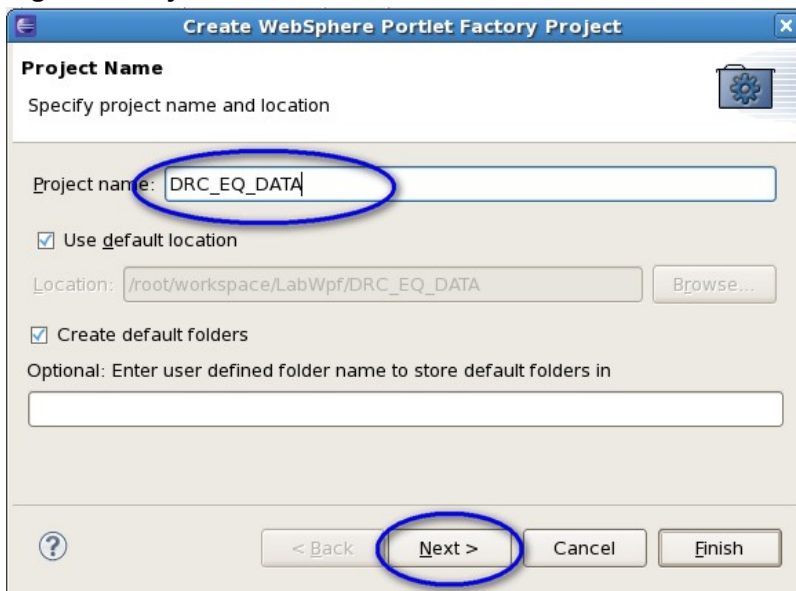
1. To begin the process of creating a new project, select File --- New --- WebSphere Portlet Factory Project (see figure 2). Alternatively, you can right-click inside the Project Explorer tab and select New --- WebSphere Portlet Factory Project from the pop-up window.

**Figure 2. WebSphere Portlet Factory Project selected**



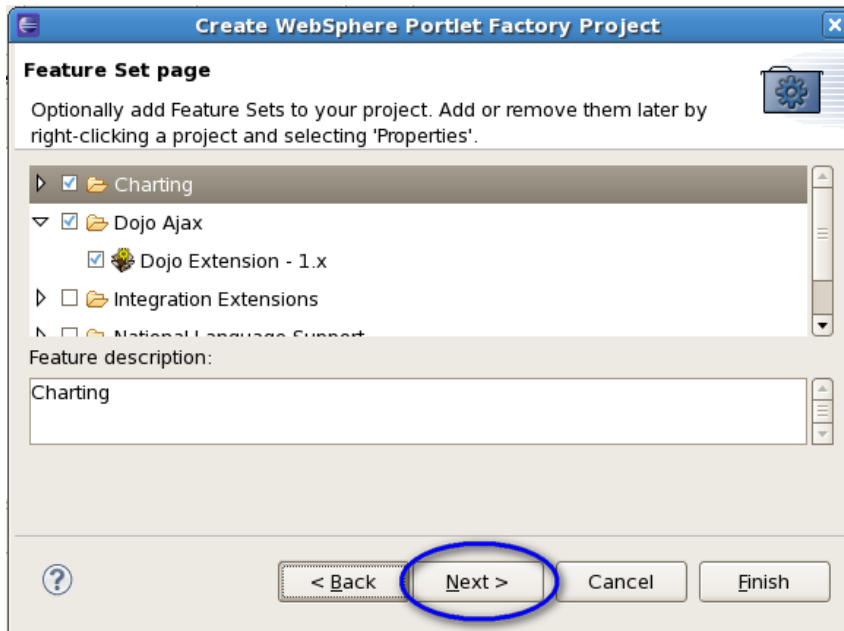
2. In the next window (see figure 3), enter “DRC\_EQ\_DATA” in the Project name field and uncheck the Create default folders options. Click Next to continue.

**Figure 3. Project Name window**



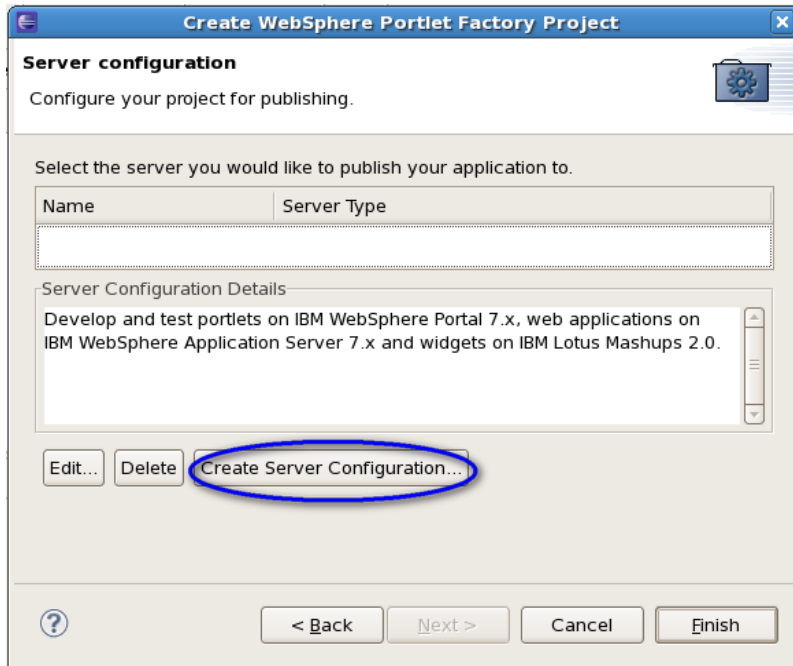
3. On the Feature Set Page window (see figure 4), simply click Next to continue. Note that, by default, Web Experience Factory adds the Charting and Dojo Ajax builders to your project.

**Figure 4. Feature Set page**



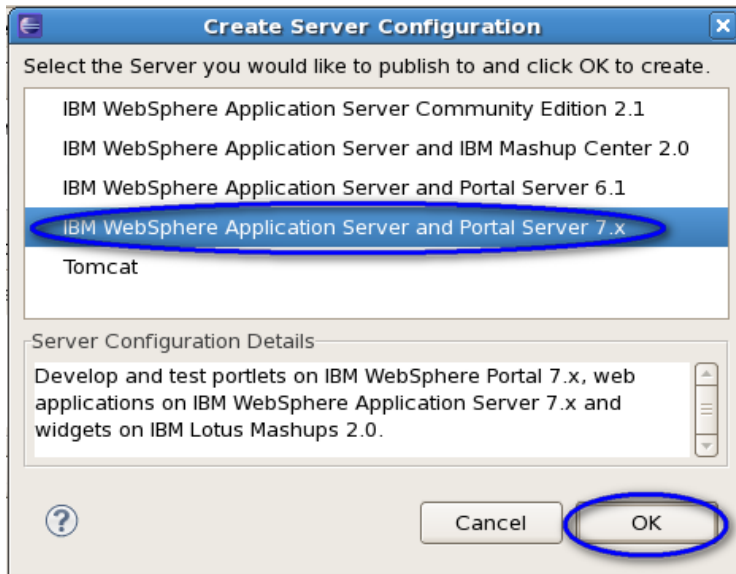
4. In the Server configuration window, click the Create Server Configuration button (see figure 5).

**Figure 5. Server configuration window**



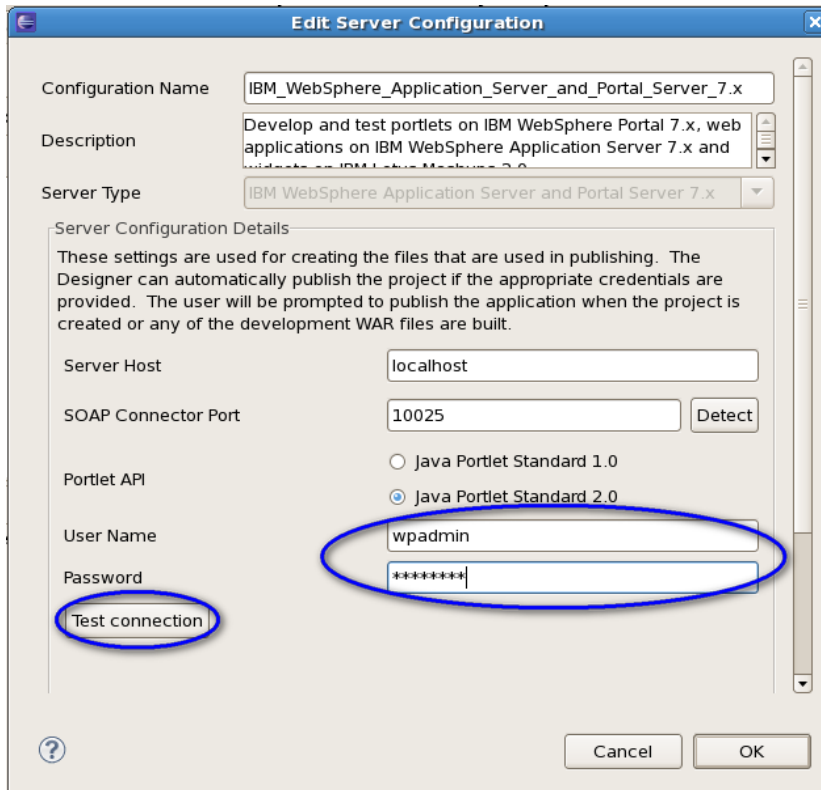
5. In the next window, select “IBM WebSphere Application Server and Portal Server 7.x”; click OK (see figure 6).

**Figure 6. Create Server Configuration window**



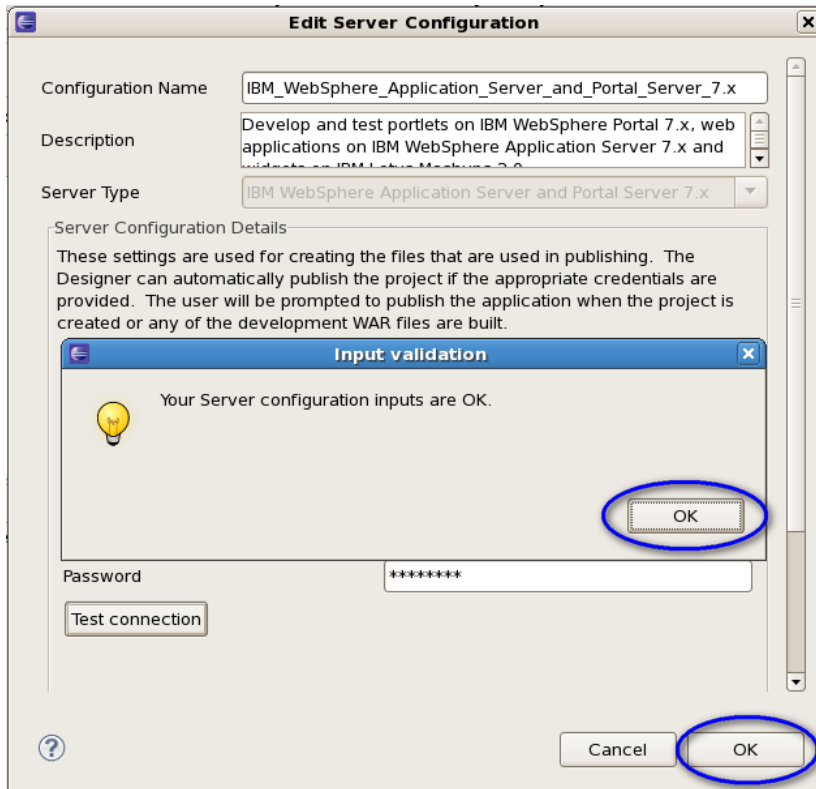
6. In the Edit Server Configuration window, enter the User Name and Password (see figure 7); click OK.

**Figure 7. Edit Server Configuration window**



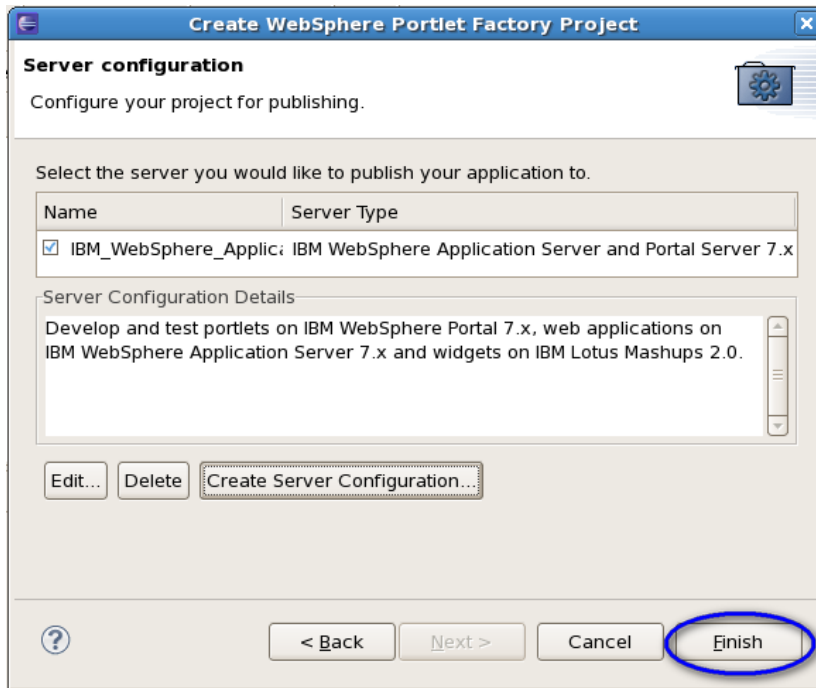
7. Click the Test connection button to ensure that your inputs are correct (see figure 8); click OK twice.

**Figure 8. Input validation pop-up**



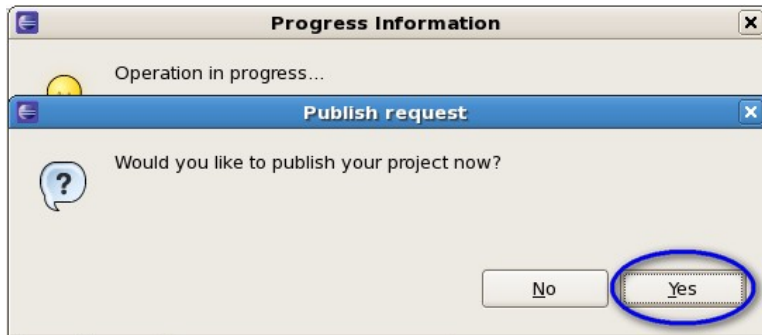
8. The Server configuration window re-displays (see figure 9); click Finish.

**Figure 9. Server configuration window**



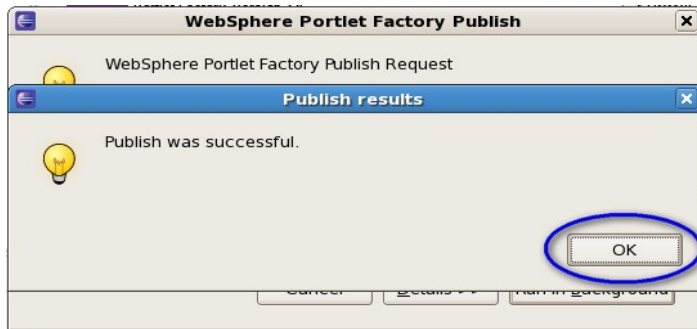
9. Click Yes in the Publish request message window (see figure 10).

**Figure 10. Publish request message box**



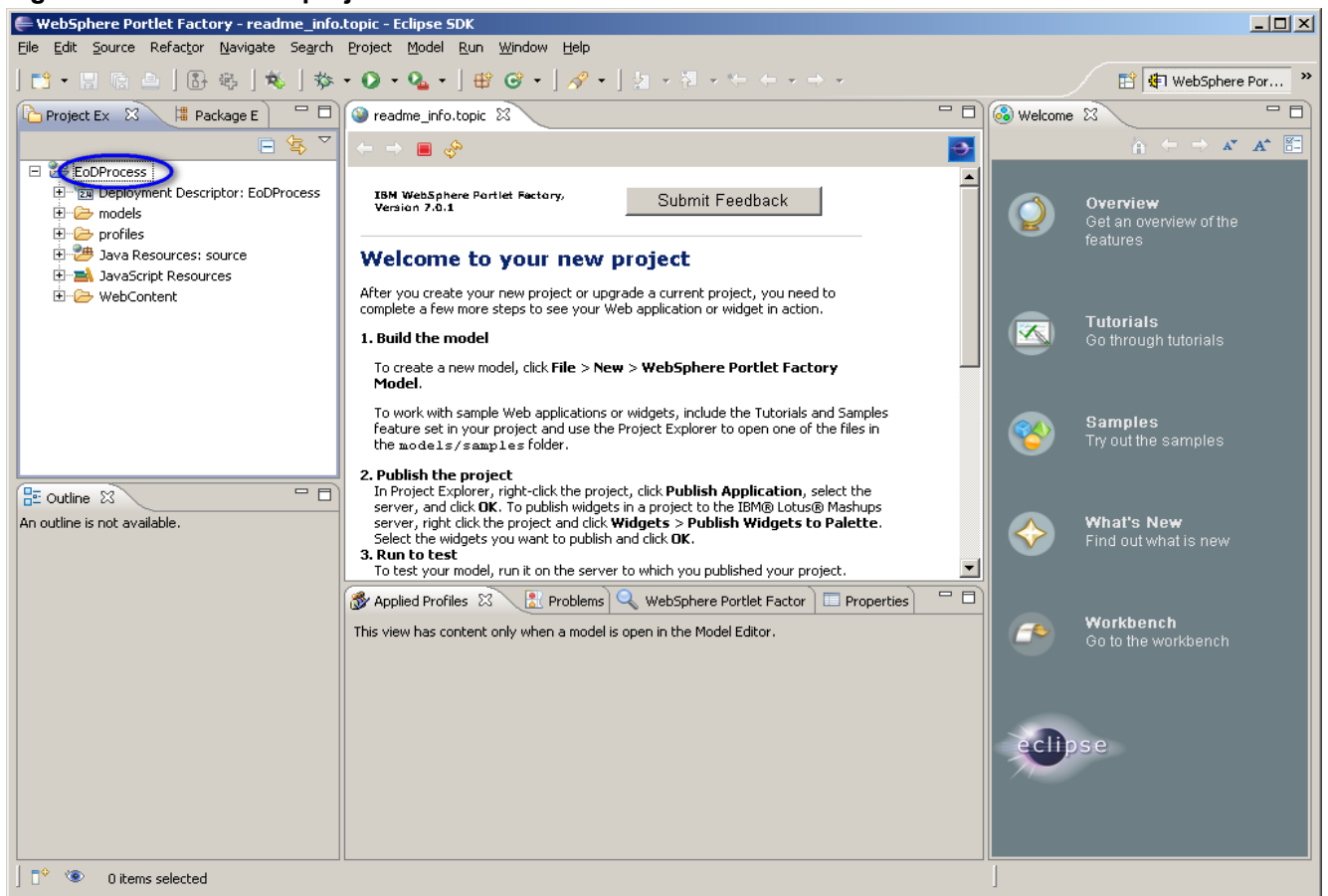
10. After you see the “Publish was successful” message, you can click OK in that message box (see figure 11).

Figure 11. Publish results window



When the project creation process is finished, you will see that a DRC\_EQ\_DATA project folder has been created (see figure 12).

Figure 12. EoDProcess project folder



## 1.2 Create a “Database Service Provider” Portlet Factory Model

Database Service Provider is a model in WPF that provides interfacing to databases and in which we can specify all queries that will be used in our application. This model works like a Data Access Object (DAO) layer in the Model View Controller (MVC) architecture.



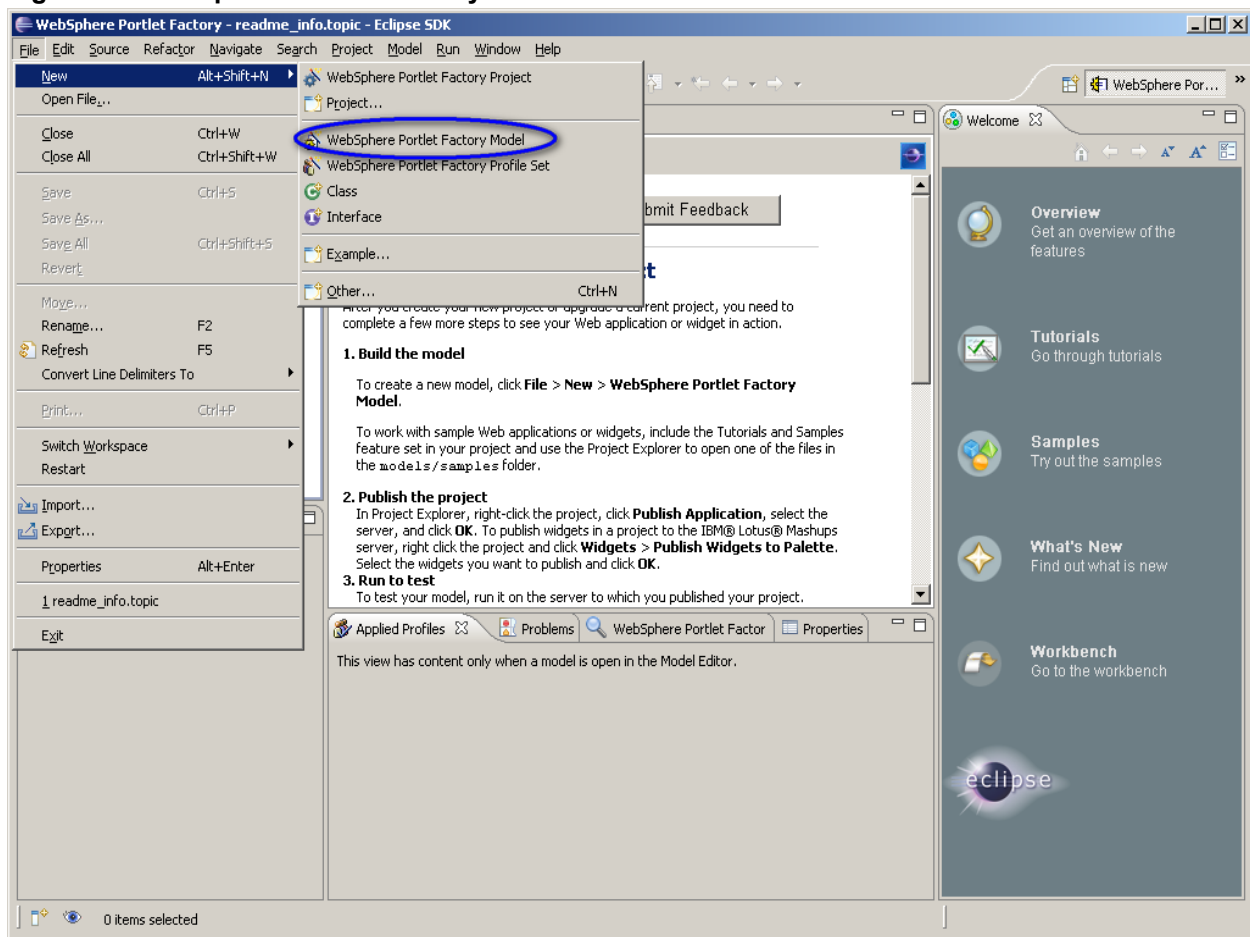
We can add other models on top of the Database Service Provider model that use services/operations generated by Database Service Provider model.

**NOTE:** As a prerequisite of this step, you must set up a Java™ Database Connectivity (JDBC) provider for your datasource. In this example we use jdbc/Bank to access our SQL Server database. The SQL script to generate our table is included below.

Here is the step-by-step process to create the Database Service Provider model used in our CRUD portlet:

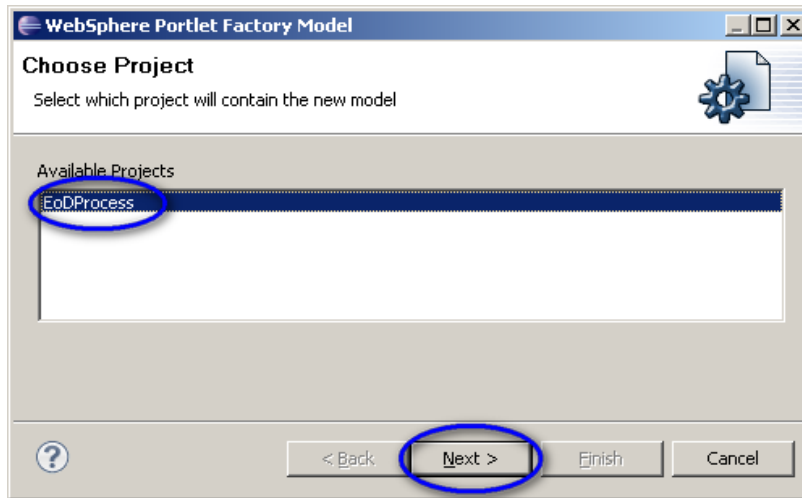
1. Create a new model for the project by selecting File --- New --- WebSphere Portlet Factory Model (see figure 13).

**Figure 13. WebSphere Portlet Factory Model selected**



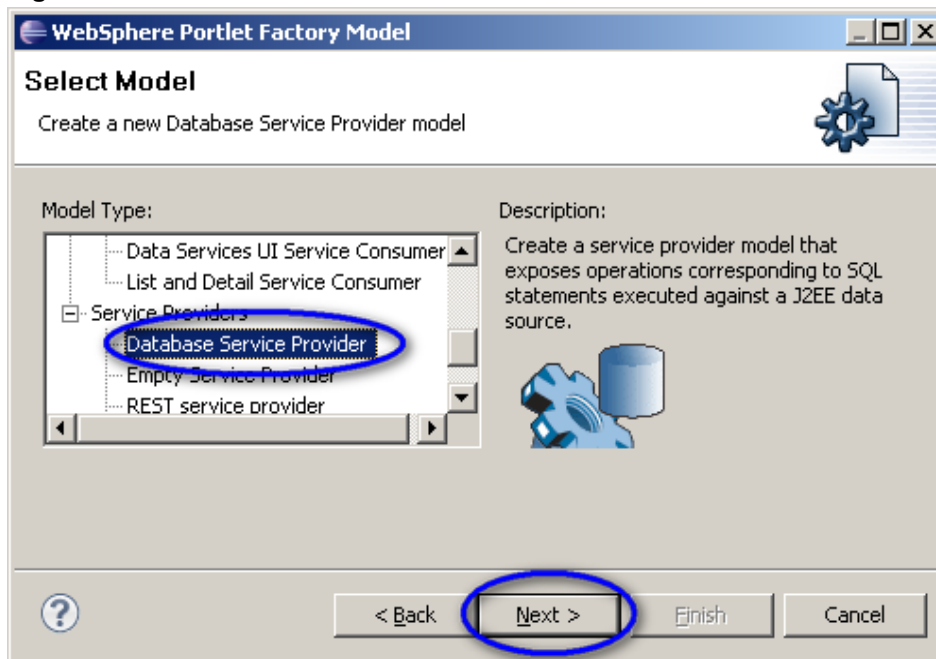
2. Select EoDProcess from the Available Projects list and click Next (see figure 14).

**Figure 14. Choose Project window**



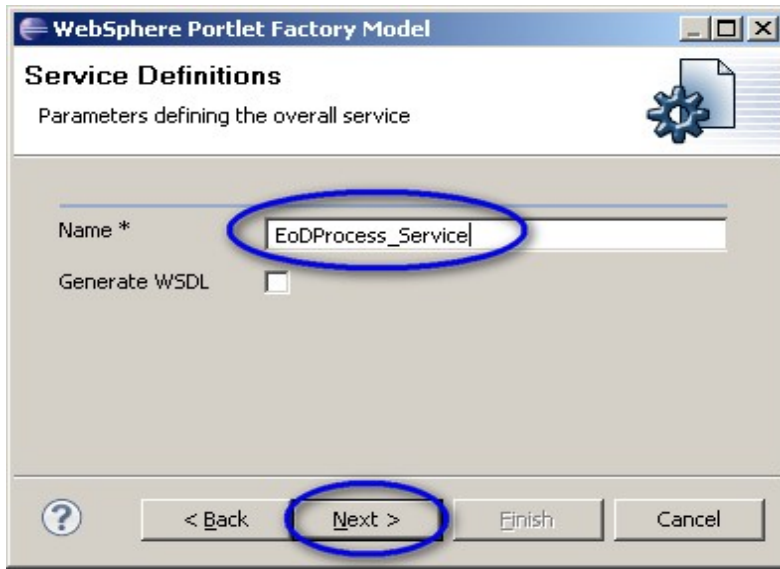
3. Select Database Service Provider for the Model Type and click Next (see figure 15).

**Figure 15. Select Model window**



4. Enter EoDProcess\_Service in the Name field and click Next (see figure 16).

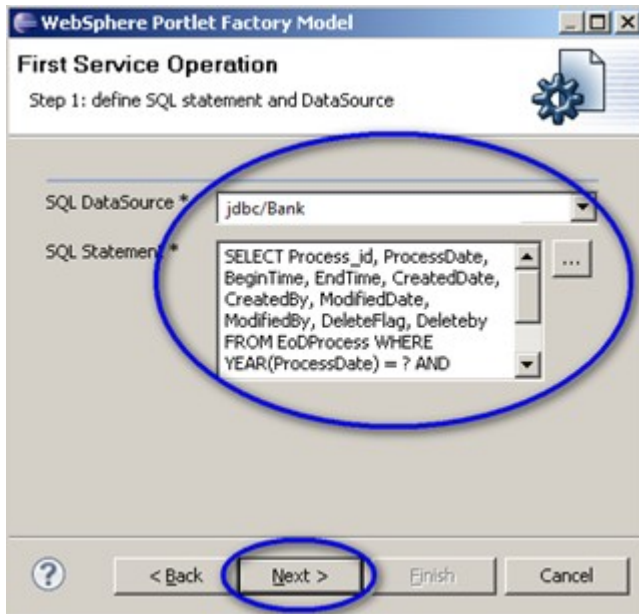
Figure 16. Service Definitions window



5. Select jdbc/Bank for the SQL DataSource drop-down list, enter the following in the SQL Statement field, and click Next (see figure 17):

```
SELECT ProcessId, ProcessDate, BeginTime, EndTime, CreatedDate,
CreatedBy, ModifiedDate, ModifiedBy, DeleteFlag, DeleteBy FROM
EoDProcess WHERE YEAR(ProcessDate) = ? AND
Convert (varchar (3),ProcessDate,100) = ? and DeleteFlag = 'false' order
by ProcessDate desc
```

Figure 17. First Service Operation (Step 1) window



6. Enter “getProcessList” for the Operation Name, and enter the other parameters as shown in figure 18; click Next.

**Figure 18. First Service Operation (Step 2) window**

WebSphere Portlet Factory Model

**First Service Operation**  
Step 2: configure service operation

Operation Name \*

Operation Inputs

Operation Input Name		
Year		↓
Month	↑	

Operation Returns

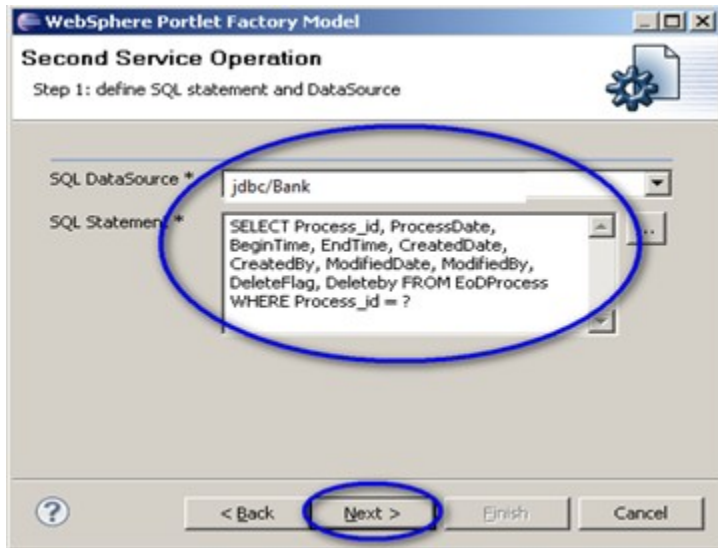
Create another operation

< Back **Next >** Finish Cancel

7. Select “jdbc/Bank” for SQL DataSource field, enter the following SQL Statement, and click Next (see figure 19):

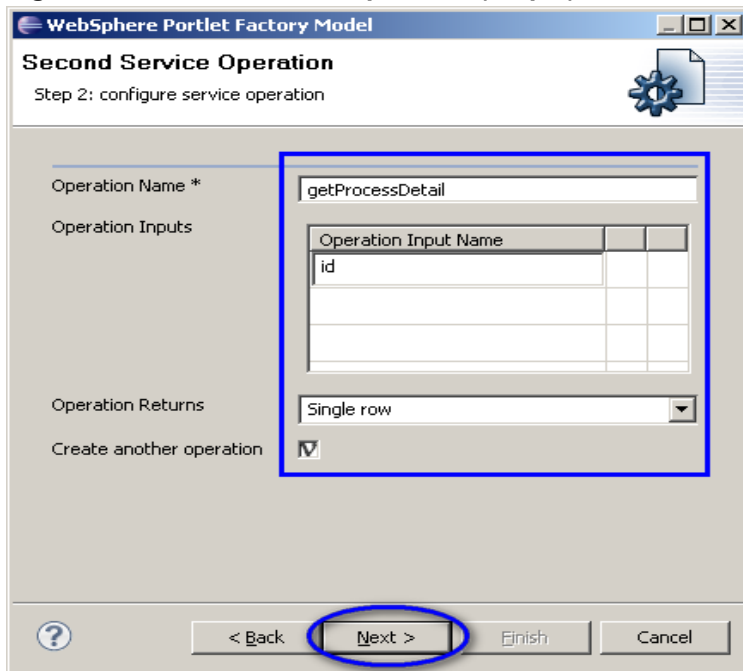
```
SELECT ProcessId, ProcessDate, BeginTime, EndTime, CreatedDate,  
CreatedBy, ModifiedDate, ModifiedBy, DeleteFlag, DeleteBy FROM  
EoDProcess WHERE ProcessId = ?
```

Figure 19. Second Service Operation (Step 1) window



8. Enter "getProcessDetail" as the Operation Name, enter the Operation parameters as shown in figure 20, and click Next.

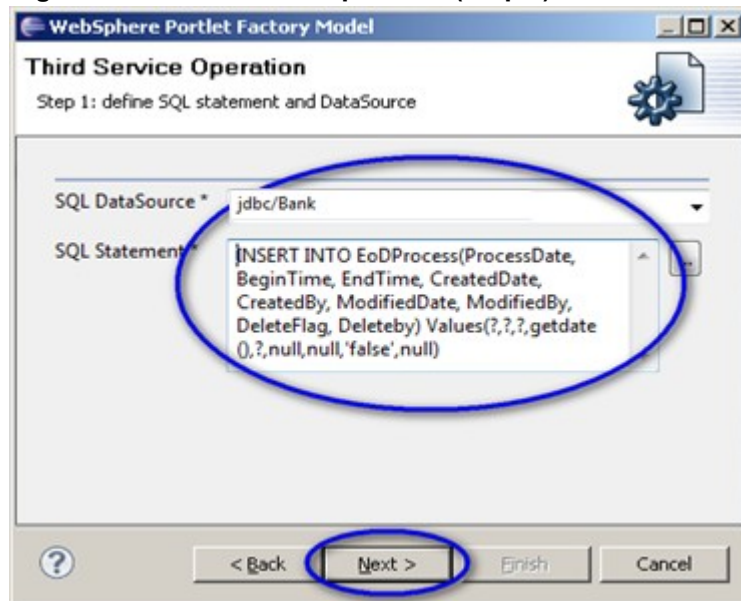
Figure 20. Second Service Operation (Step 2) window



9. Select "jdbc/Bank" for SQL DataSource field, enter the following SQL Statement, and click Next (see figure 21):

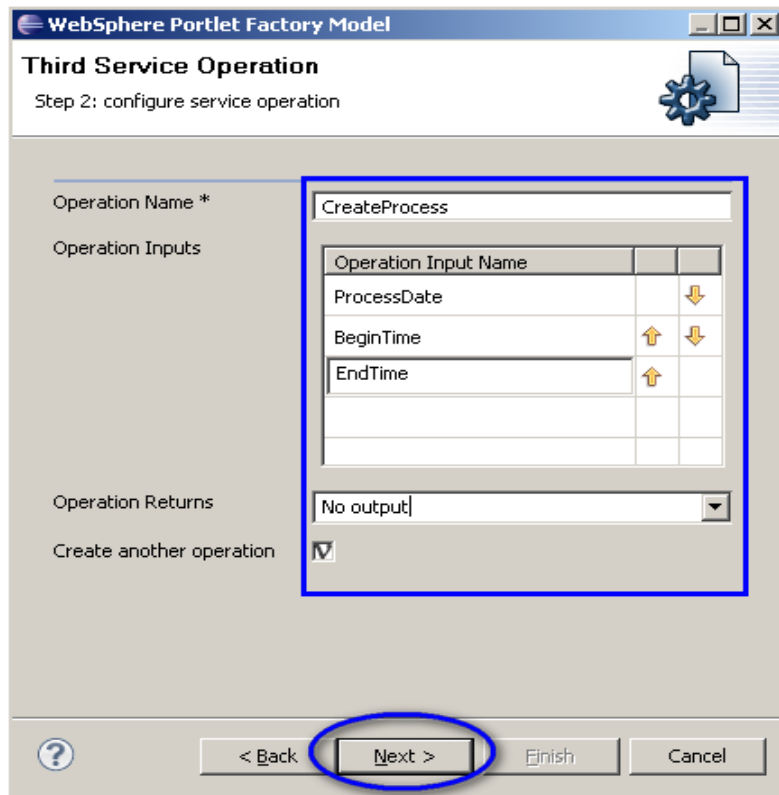
```
INSERT INTO EoDProcess(ProcessDate, BeginTime, EndTime, CreatedDate,  
CreatedBy, ModifiedDate, ModifiedBy, DeleteFlag, DeleteBy)  
Values(?,?,?,getdate(),?,null,null,'false',null)
```

**Figure 21. Third Service Operation (Step 1) window**



10. Enter "CreateProcess" in the Operation Name field, enter the parameters as shown in figure 22, and click Next.

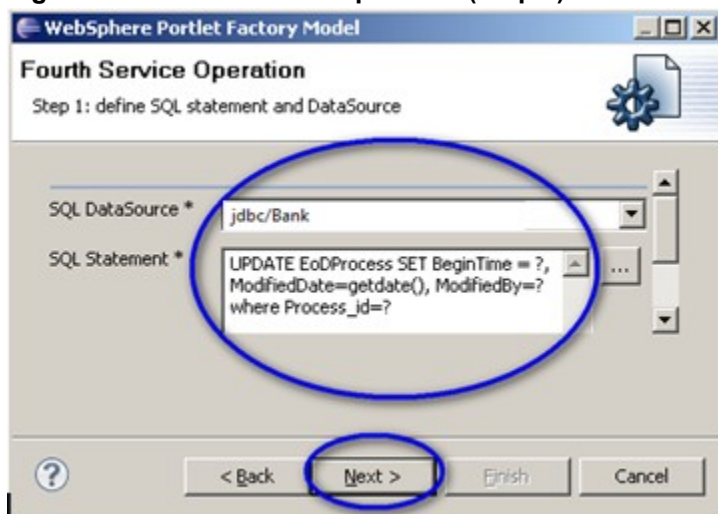
Figure 22. Third Service Operation (Step 2) window



11. Select "jdbc/Bank" for the SQL DataSource field, enter the following SQL Statement, and click Next (see figure 23):

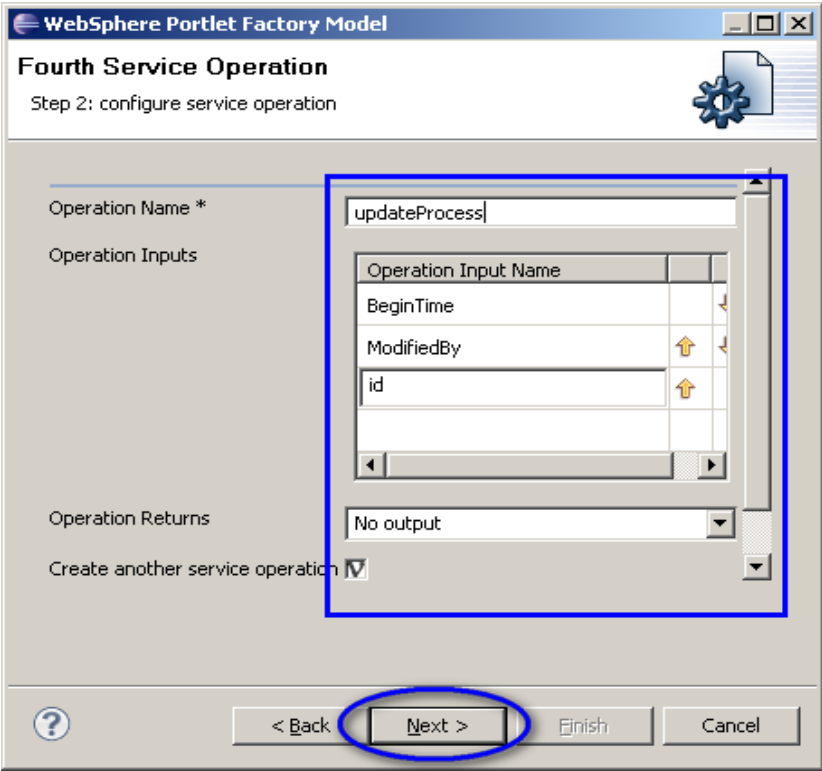
```
UPDATE EoDProcess SET BeginTime = ?, ModifiedDate=getdate(), ModifiedBy=?  
where ProcessId=?
```

Figure 23. Fourth Service Operation (Step 1) window



12. Enter “updateProcess” in the Operation Name field, enter the parameters as shown in figure 24, and click Next.

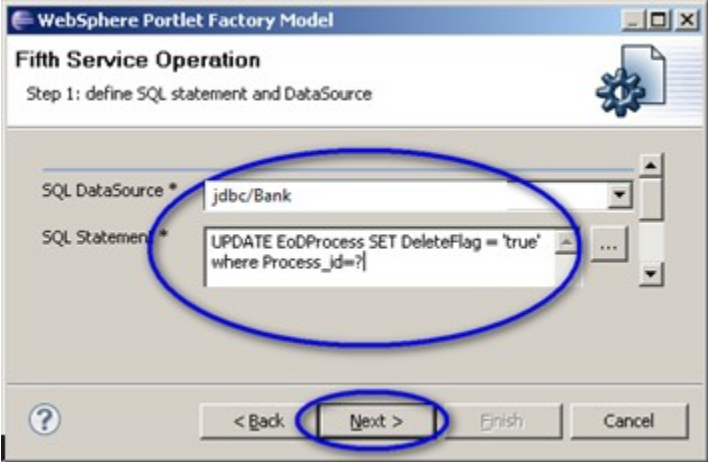
Figure 24. Fourth Service Operation (Step 2) window



13. Select “jdbc/Bank” for the SQL DataSource field, enter the following SQL Statement, and click Next (see figure 25):

```
UPDATE EoDProcess SET DeleteFlag = 'true' where ProcessId=?
```

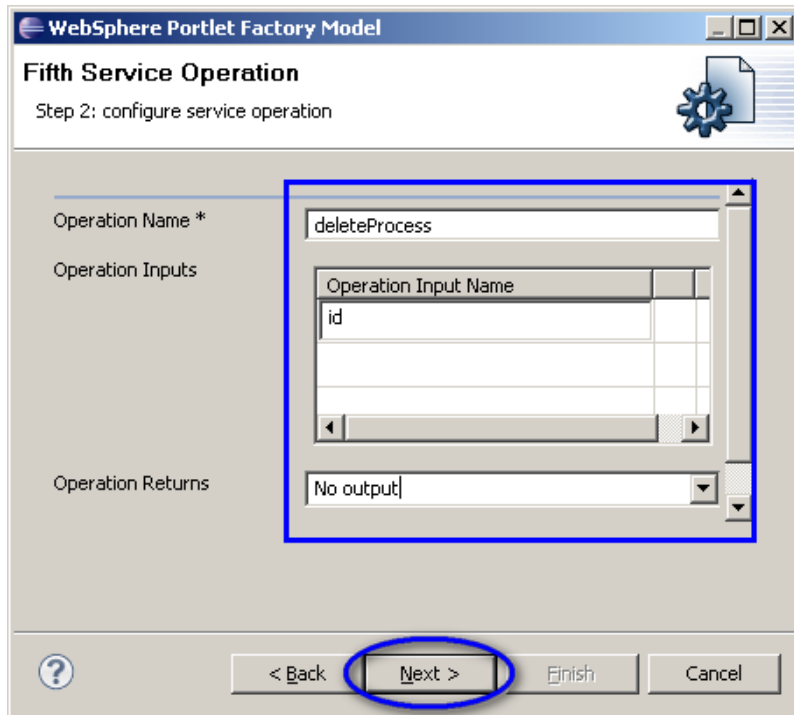
Figure 25. Fifth Service Operation (Step 1) window





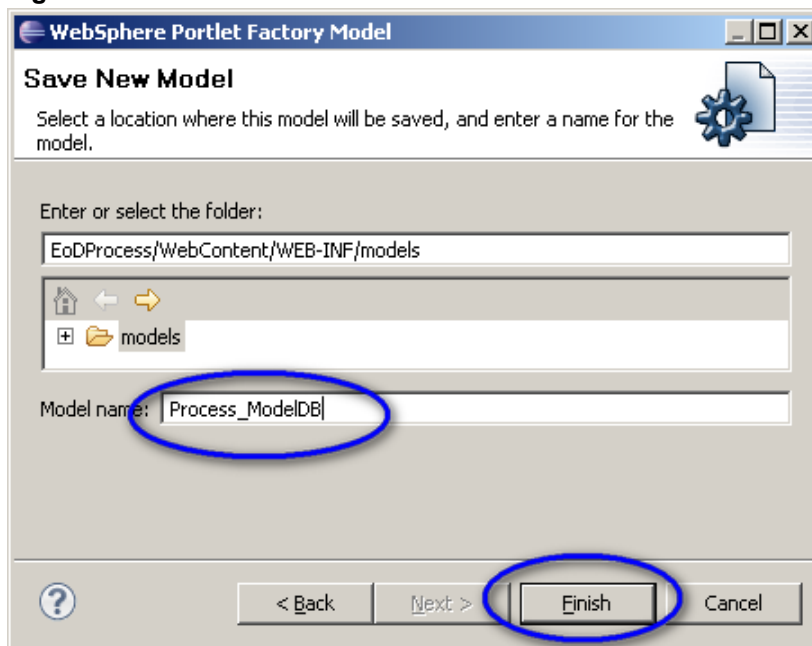
14. Enter “deleteProcess” in the Operation Name field, enter the parameters as shown in figure 26, and click Next.

**Figure 26. Fifth Service Operation (Step 2) window**



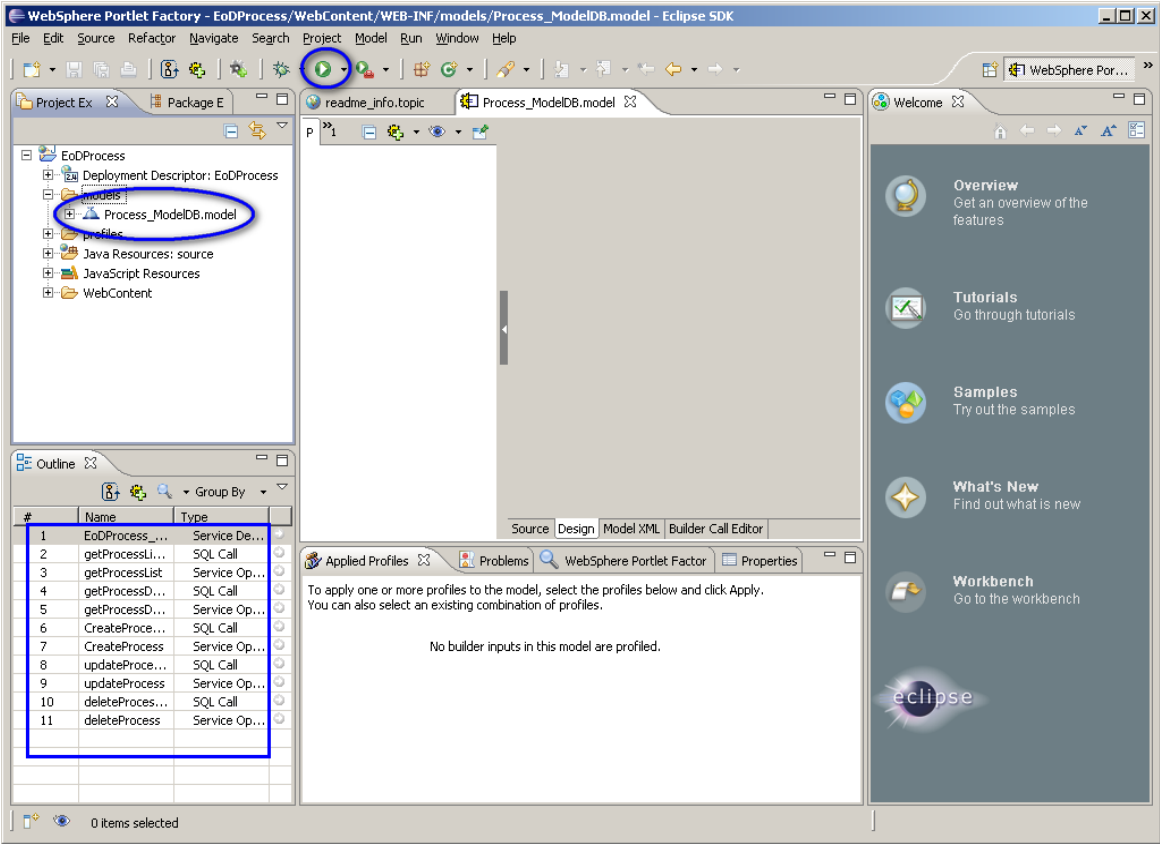
15. Name the model “Process\_ModelIDB” and click Finish (see figure 27).

**Figure 27. Save New Model window**



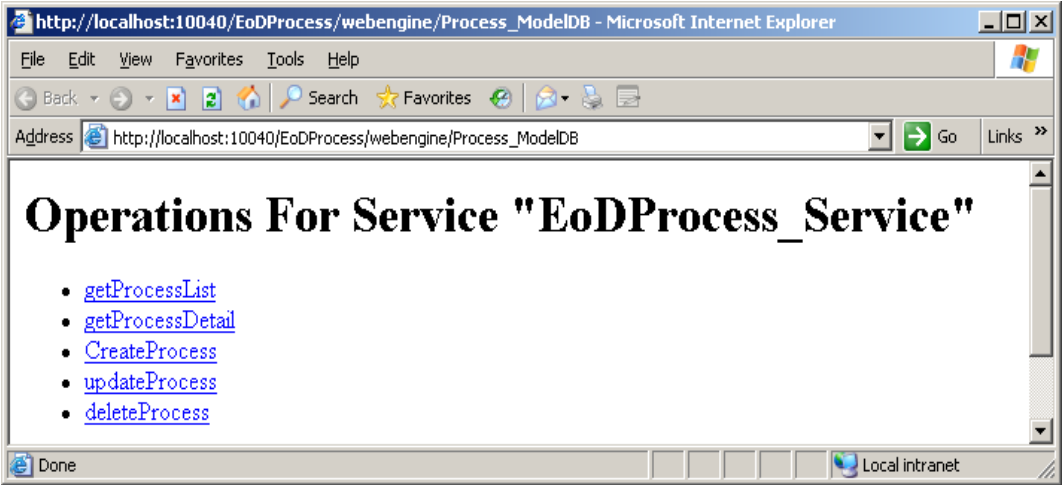
16. When the model creation process is finished, you will see that a Process\_ModelDB model has been created (see figure 28).

Figure 28. Newly created DRC\_EQ\_ModelIDB model



17. To run the model, click the green arrow button at the top of the page. The window in figure 29 displays.

Figure 29. Run the model



18. Click the “getProcessList” link in the above window; the windows in figures 30 and 31 display.

Figure 30. Page Filter getProcessList link

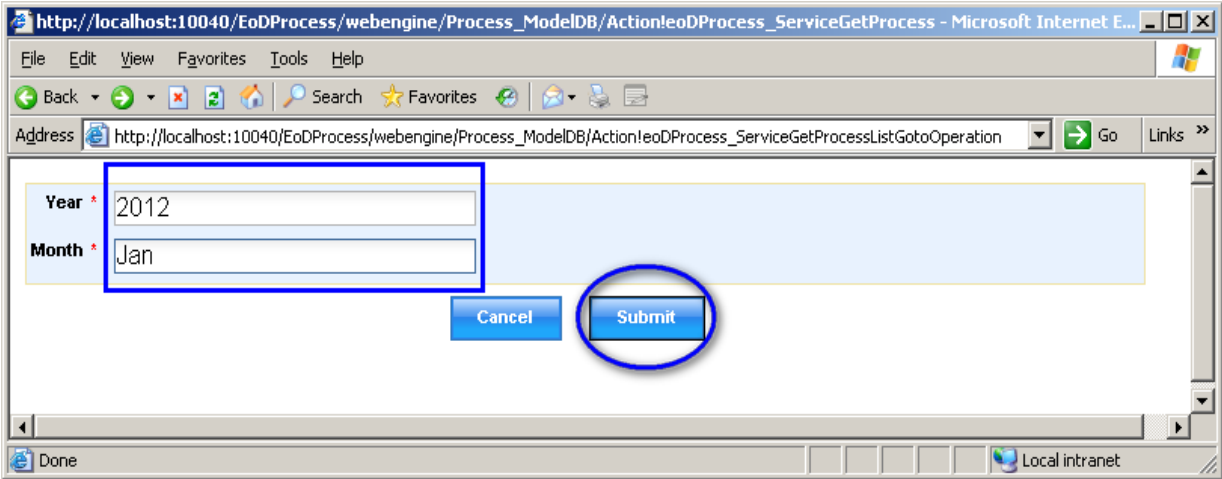
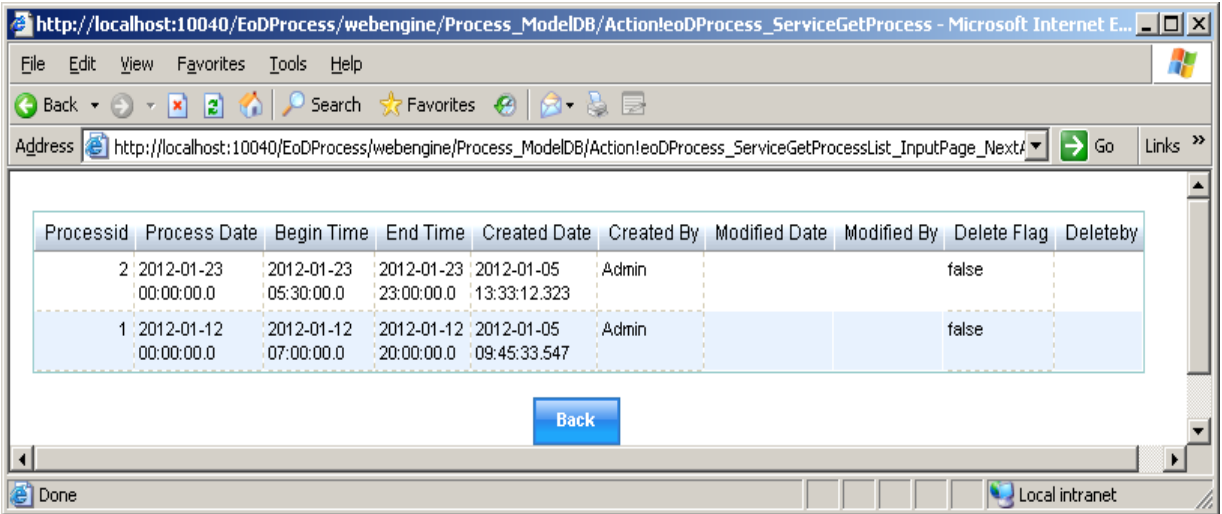
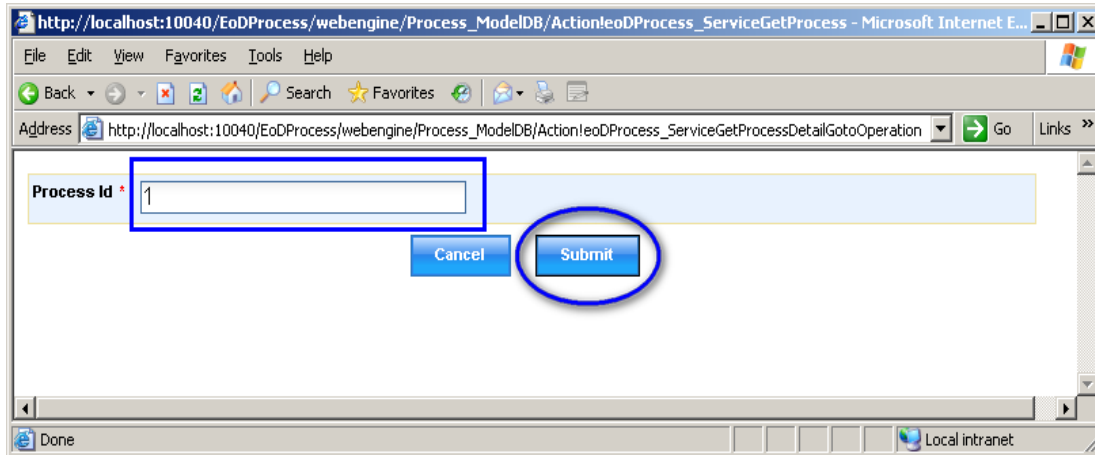


Figure 31. Page Result getProcessList link

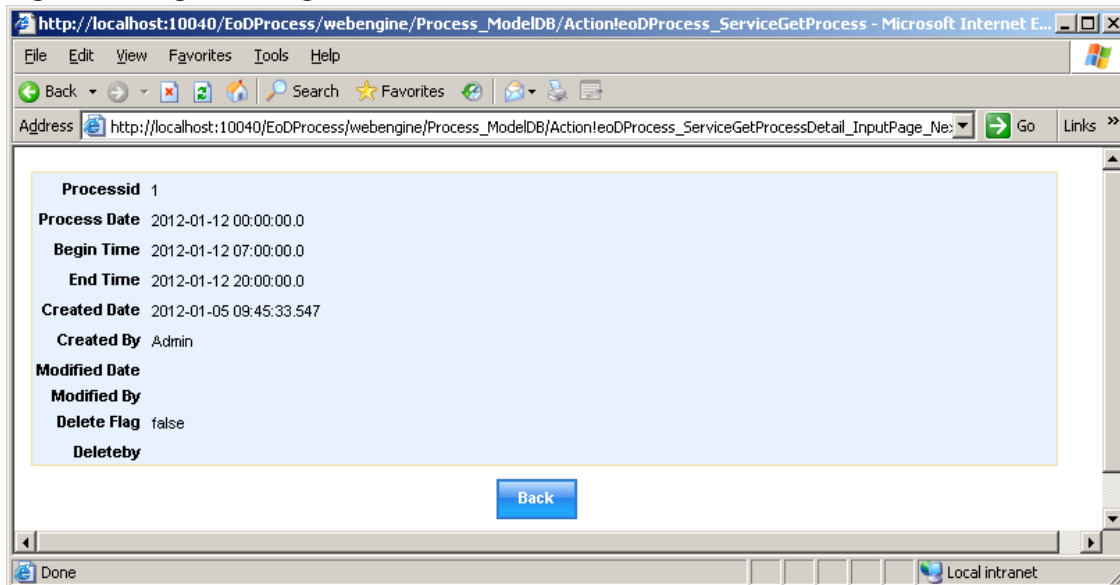


19. Click the “getProcessDetail link; the windows in figures 32 and 33 display.

**Figure 32. Page Filter getProcessDetail link**

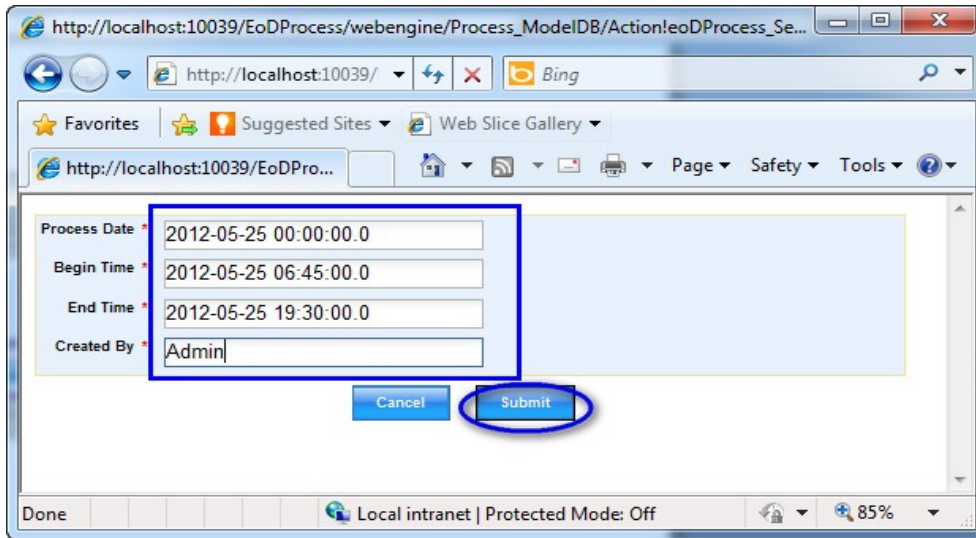


**Figure 33. Page Result getProcessDetail link**



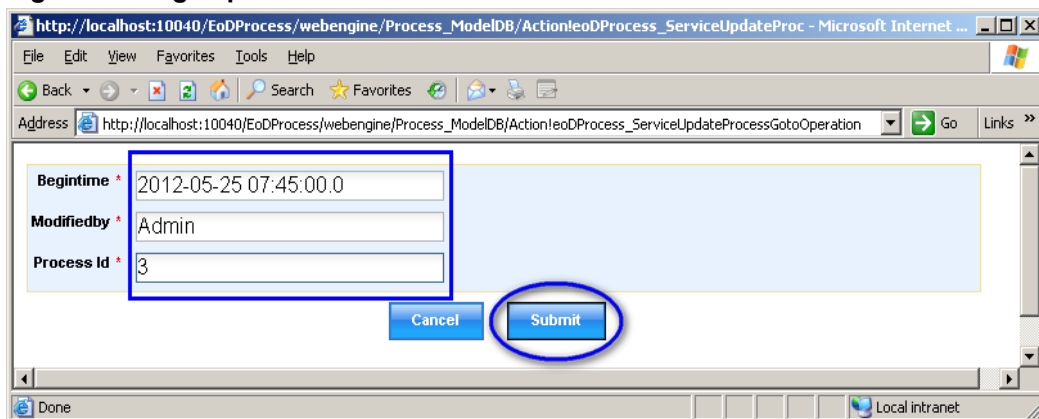
20. Click the CreateProcess link; the window in figure 34 displays.

**Figure 34. Page CreateProcess link**



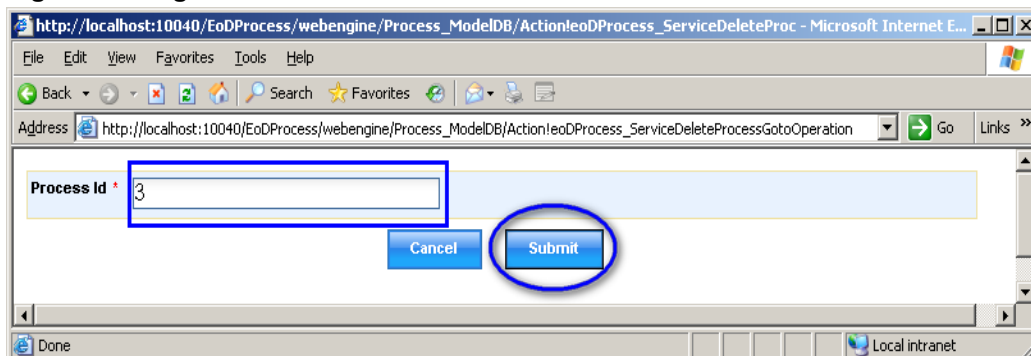
21. Click the updateProcess link; the window in figure 35 displays.

**Figure 35. Page updateProcess Link**



22. Click deleteProcess; the window in figure 36 displays.

**Figure 36. Page deleteProcess link**



23. Finally, test all database services/operations that we have defined above, ensuring there are no errors. We will use the DAO-like service/operation in a later model.

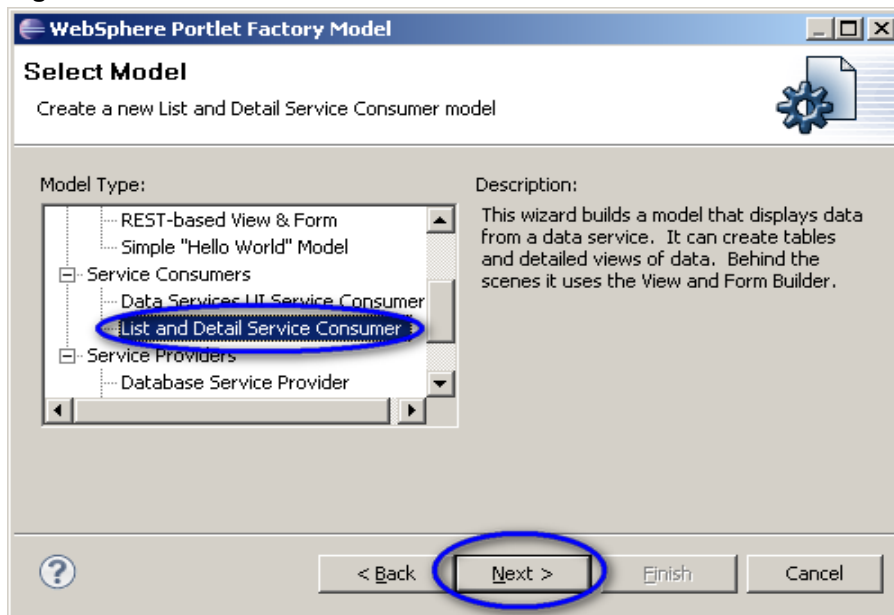
### 1.3 Create a “List and Detail Service Consumer” WPF Model

In this, the main model of our CRUD portlet, we can specify the actions/functions provided in our application; in this case, our actions are Create, Retrieve, Update, and Delete (CRUD). We also specify the flow of each action in this model.

To create a new model for the project:

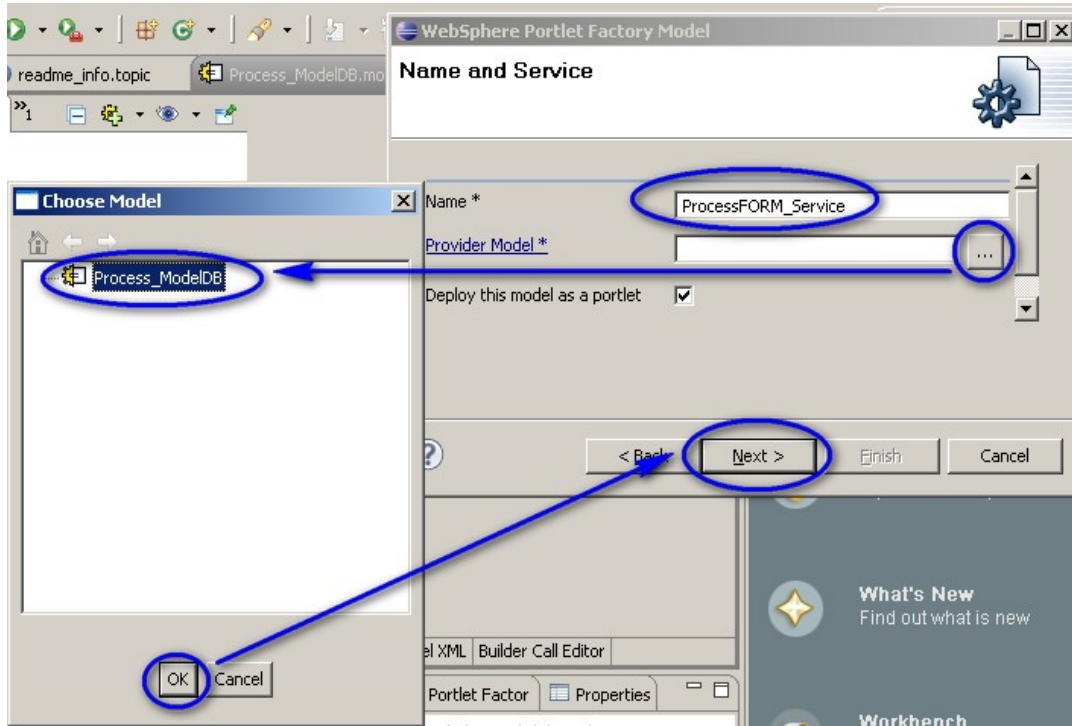
1. Select File -- New -- WebSphere Portlet Factory Model.
2. Choose the Project as EoDProcess; click Next.
3. Select “List and Detail Service Consumer” as the Model type, and click Next (see figure 37).

Figure 37. Select Model window



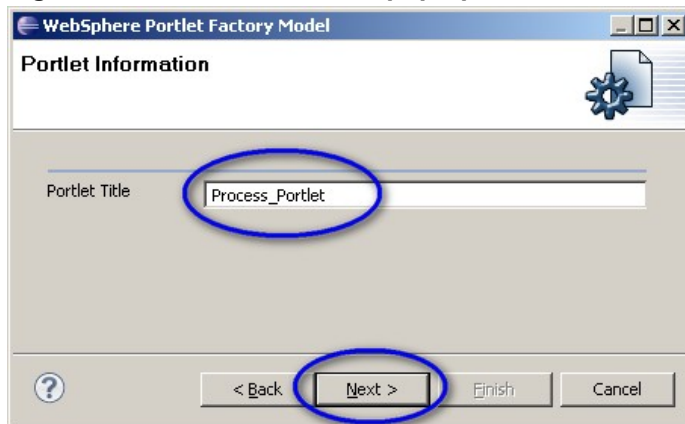
4. In the Name and Service pop-up window (see figure 38), enter “ProcessFORM\_Service” in the Name field, select Process\_ModelDB as the Provider Model, and click Next.

**Figure 38. Name and Service pop-up window**



5. In the portlet Information pop-up window (see figure 39), enter “Process\_Portlet” in the portlet title field, then click Next.

**Figure 39. Portlet Information pop-up window**



6. In the Main View window (see figure 40), enter the following values for these parameters, then click Next:
  - a) View Data Operation: getProcessList
  - b) Input Values:

**Year:**

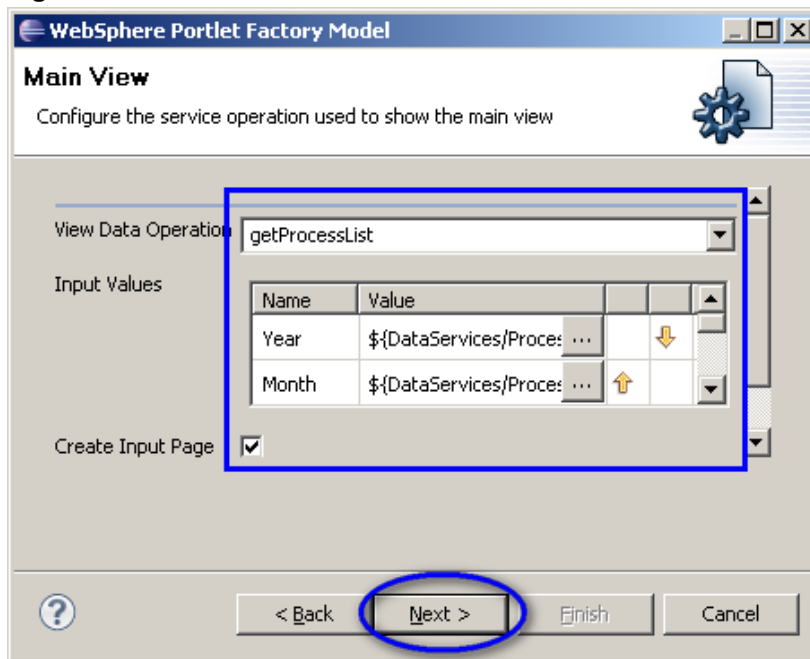
`${DataServices/ProcessFORM_Service/getProcessList/inputs/getProcessListQueryInputs/Year}`

**Month:**

`${DataServices/ProcessFORM_Service/getProcessList/inputs/getProcessListQueryInputs/Month}`

- c) Enable (check mark) the Create Input Page option.

**Figure 40. Main View window**

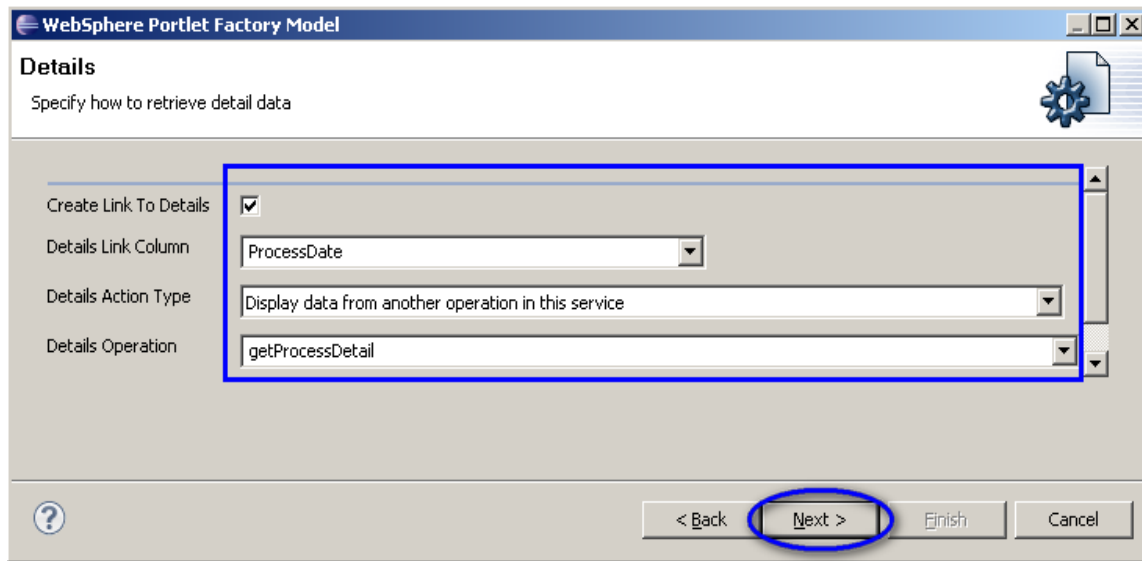


- 7. In the Detail window (see figure 41), enable the “Create Link To Details” option, and enter the following values for these parameters; click Next:

- a) Details Link Column: ProcessDate
- b) Details Action Type: Display data from another operation in this service
- c) Details Operation: getProcessDetail



**Figure 41. Details window**

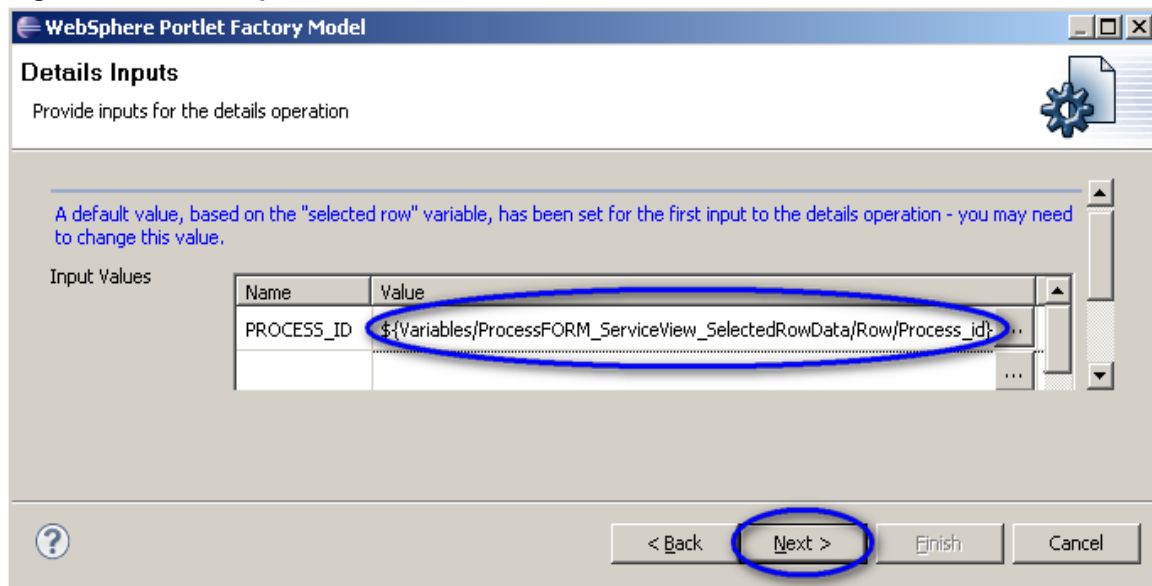


8. In the Detail Inputs window, select

`${Variables/ProcessFORM_ServiceView_SelectedRowData/Row/Process_id}`

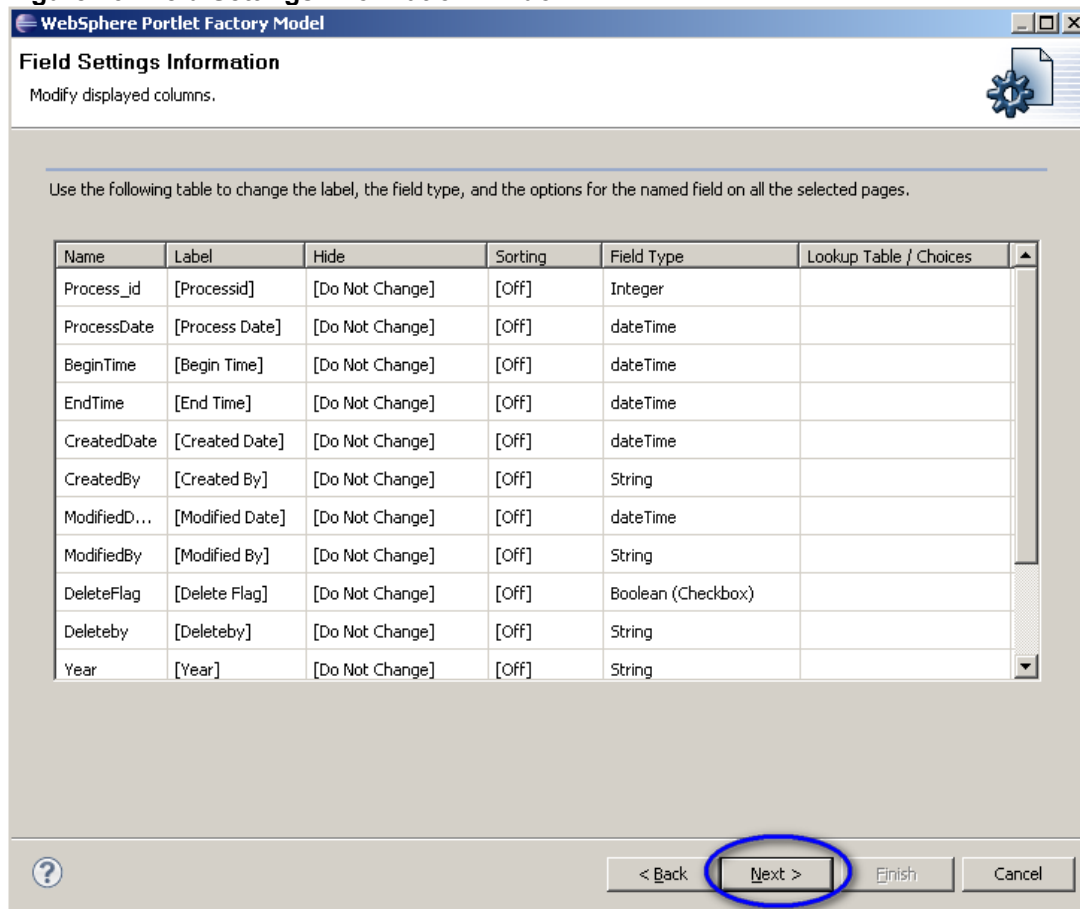
for input Values Process\_ID and click Next (see figure 42).

**Figure 42. Details Inputs window**



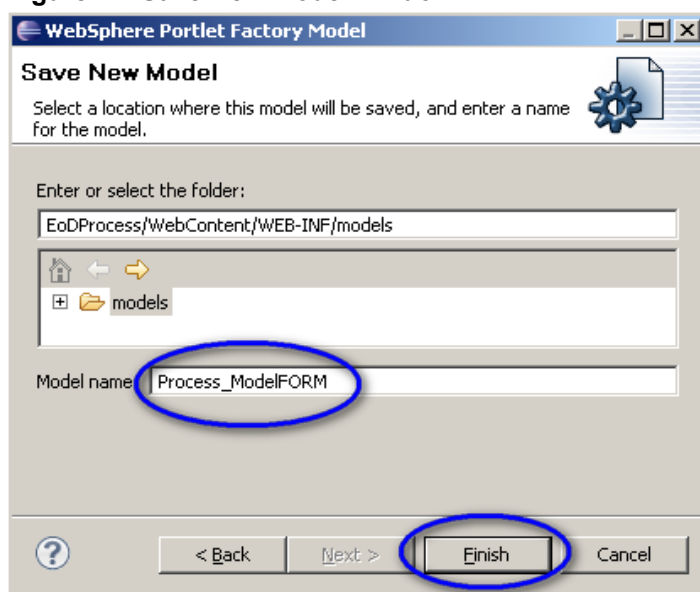
9. In the Field Settings Information window, click Next (see figure 43).

**Figure 43. Field Settings Information window**



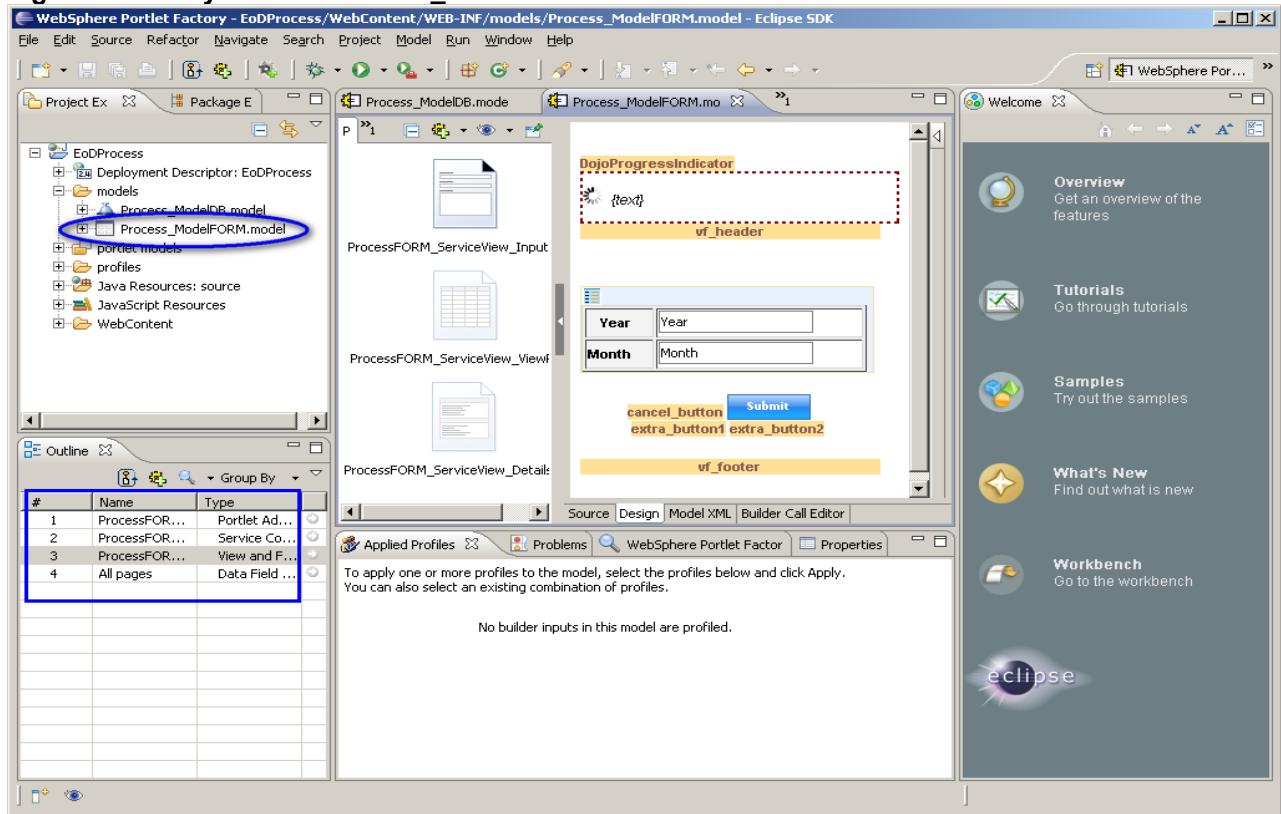
10. Enter "Process\_ModelFORM" in the Model name field and click Finish (see figure 44).

**Figure 44. Save New Model window**



When the model creation process is finished, you will see that a Process\_ModelFORM model has been created (see figure 45).

**Figure 45. Newly created Process\_ModelFORM model**



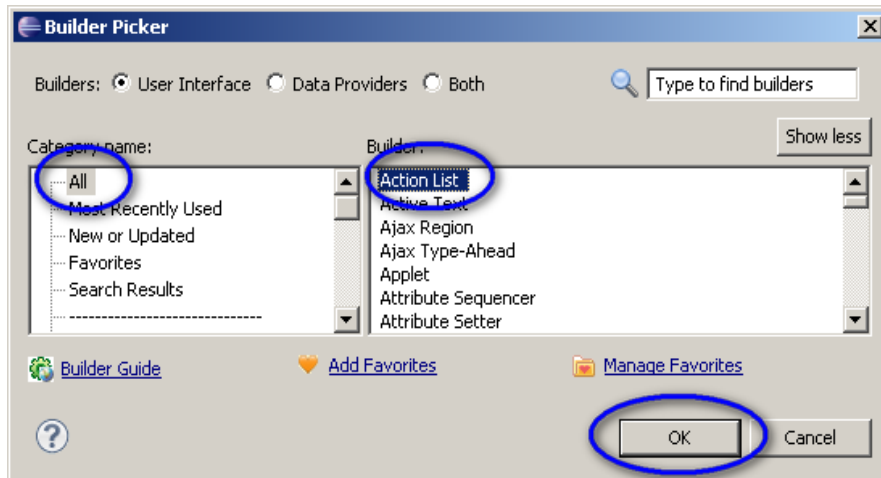
## 1.4 Create Form

The model we've just created is basic. To add more functionalities and features to it, we need to add Builders to the model.

### 1.4.1 Add Action List Builder

1. In the Outline view, click the Builder picker icon, select All, and then Action List from the Builder from the list; click OK (see figure 46).

**Figure 46. Action List Builder selected**

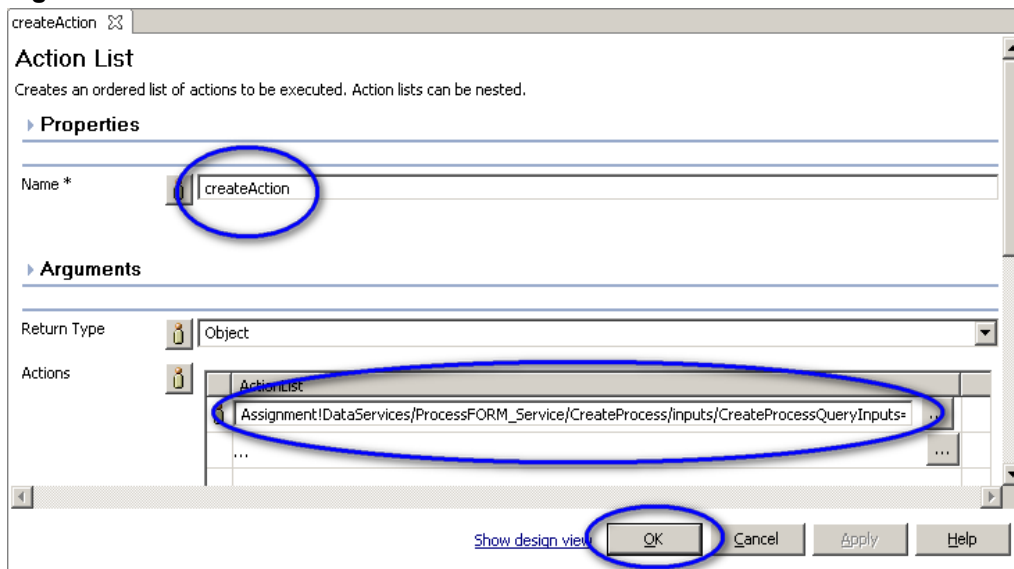


2. Enter createAction in the Action List Name field, and enter

AssignmentDataServices/ProcessFORM\_Service/CreateProcess/inputs/CreateProcess  
QueryInputs=

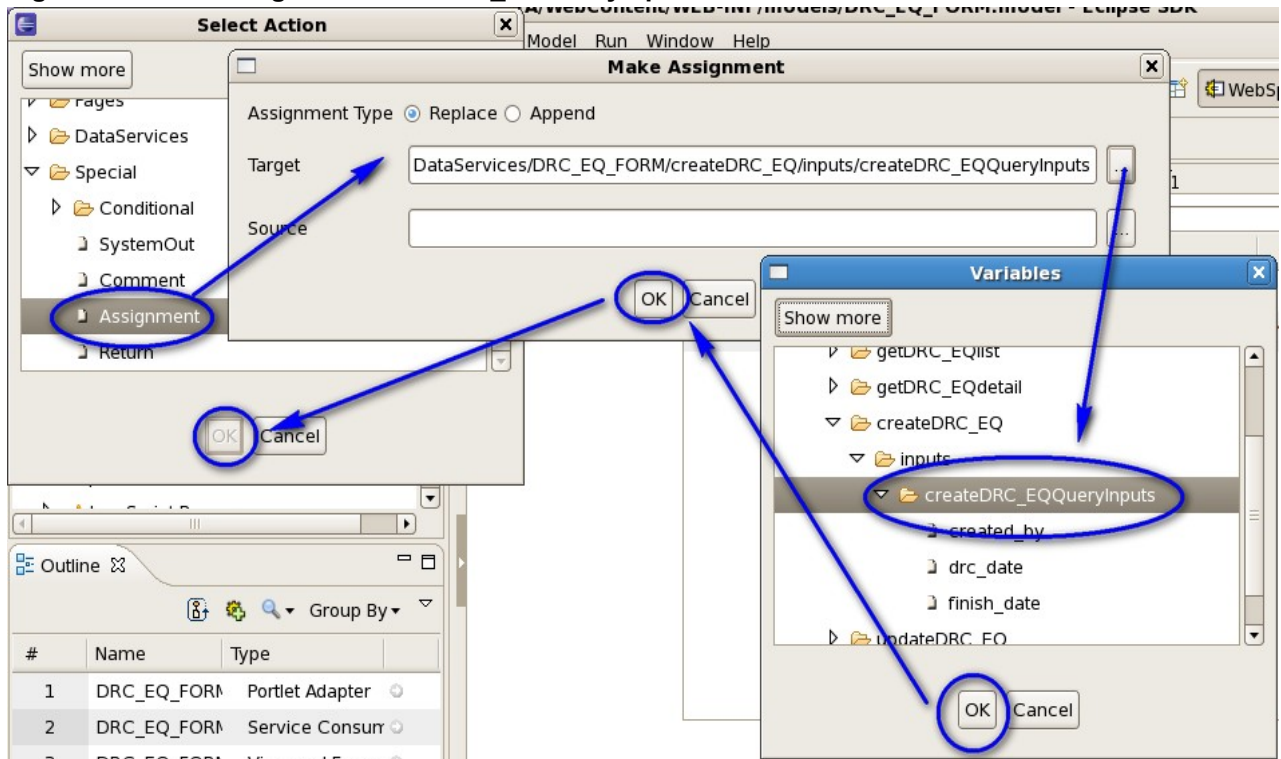
in the Actions values, then click OK (see figure 47).

**Figure 47. Action List window**



3. Select the Assignment action and, in the Make Assignment window, select the Target as createDRC\_EQQueryinputs; click OK (see figure 38).

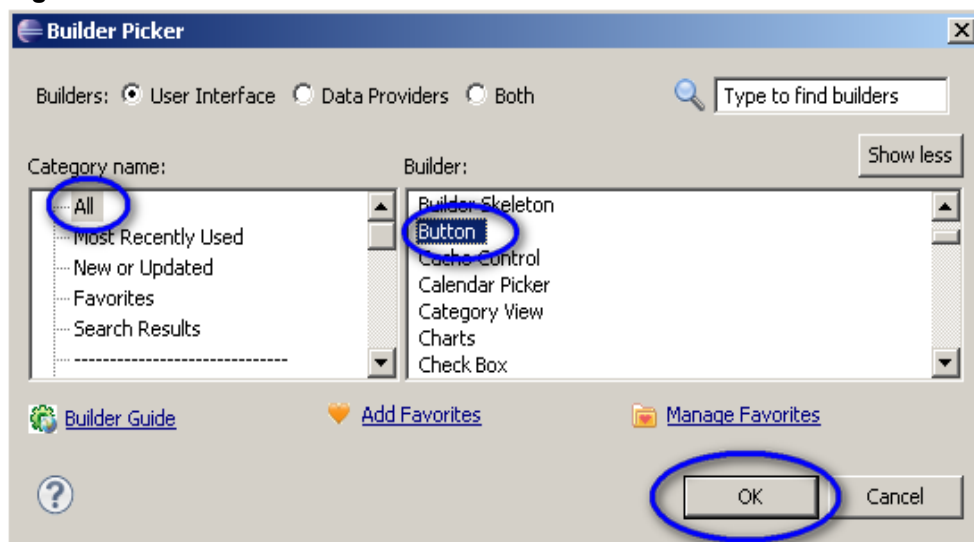
**Figure 38. Create Target as createDRC\_EQQueryinputs**



### 1.4.2 Add Button Builder

1. In the Outline view, click the builder picker icon, select All, and then "Button" from the Builder list; click OK (see figure 48).

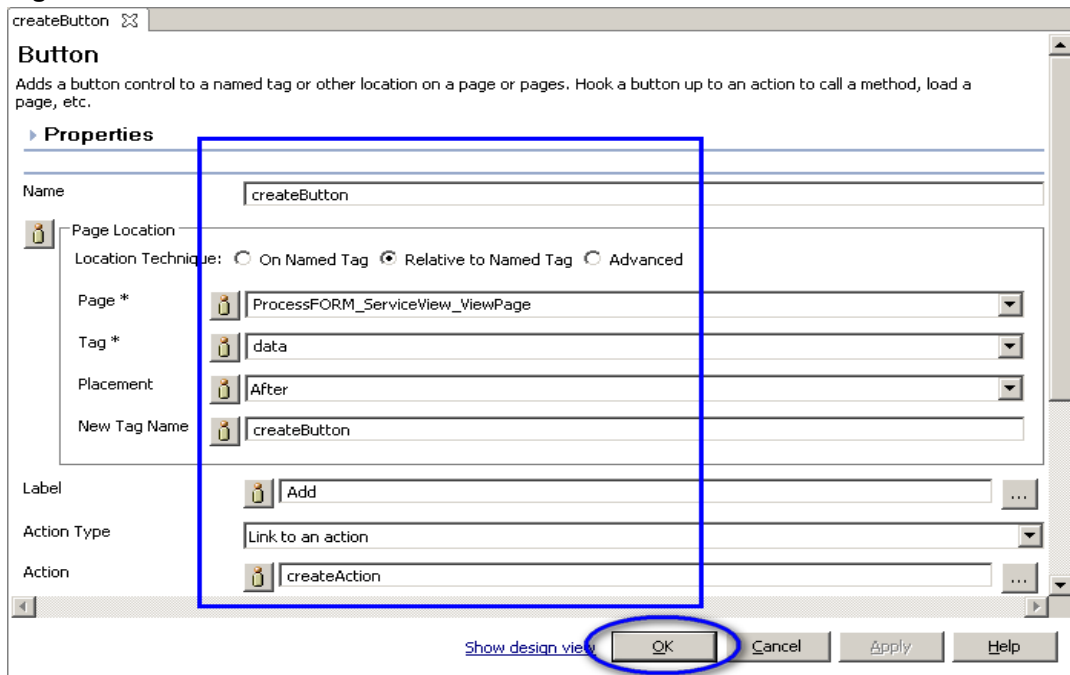
**Figure 48. Button builder selected**



2. Enter createButton for the Button Name, select the “Relative to Named Tag” option for the Location Technique, and add the following inputs for these parameters (see figure 49):

- Page: ProcessFORM\_ServiceView\_ViewPage
- Tag: data
- Placement: After
- New Tag Name: createButton
- Label: Add
- Action Type: Link to an action
- Action: createAction

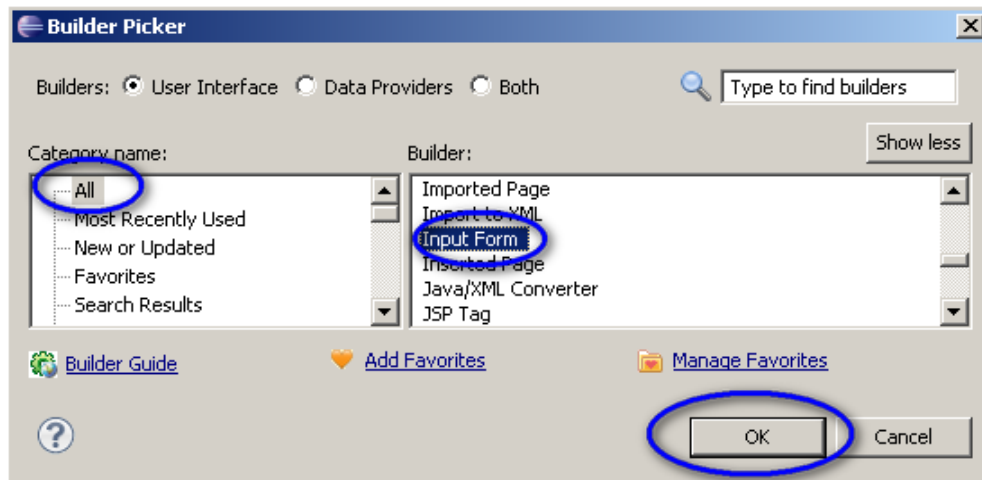
**Figure 49. Button window**



### 1.4.3 Add Input Form Builder

1. In the Outline view, click the Builder picker icon, select All and then “Input Form” from the Builder list; click OK (see figure 50).

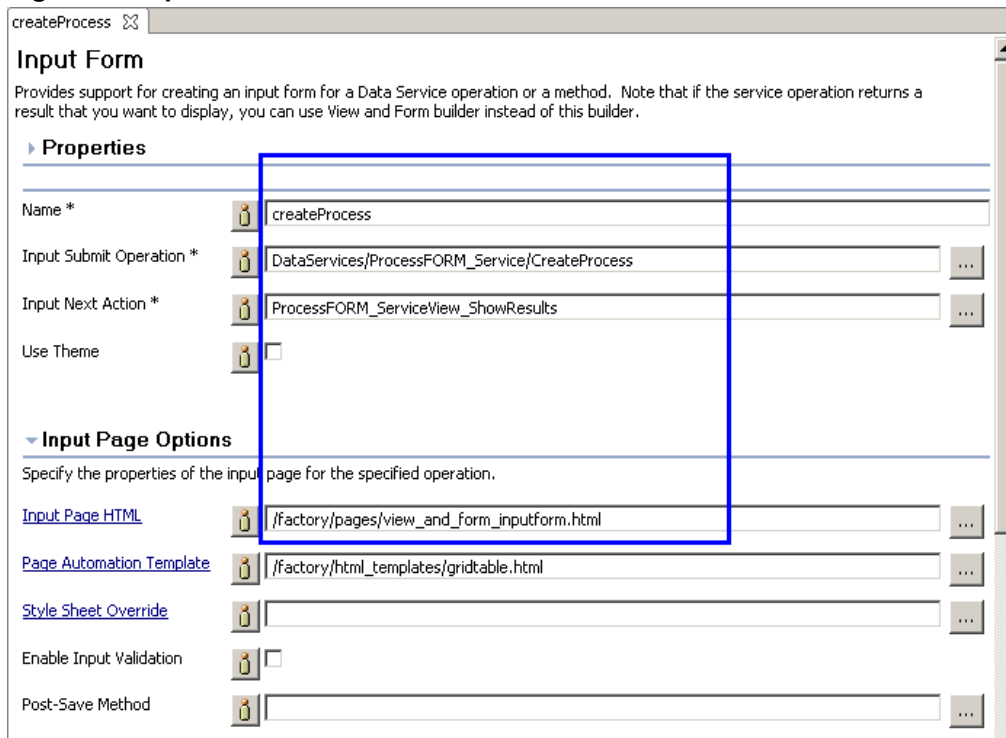
**Figure 50. Input Form builder selected**

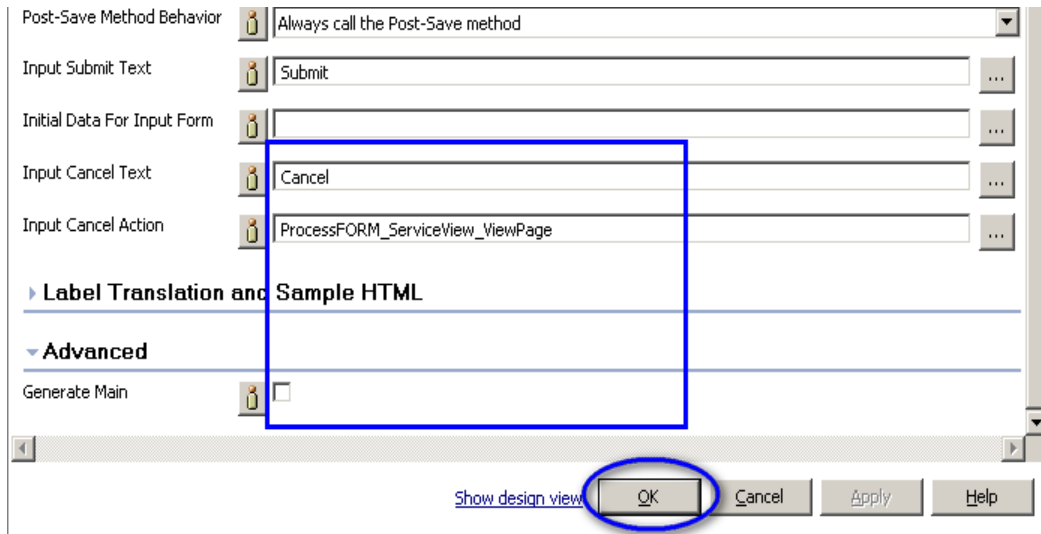


2. Enter “createProcess” for the Input Form Name, uncheck the Use Theme option, and add the following inputs for these parameters (see figure 51):

- Input Submit Operation: DataServices/ProcessFORM\_Service/createProcess
- Input Next Action: ProcessFORM\_ServiceView\_ShowResults
- Input Page HTML: /factory/pages/view\_and\_form\_inputform.html
- Input Cancel Text: Cancel
- Input Cancel Action: ProcessFORM\_ServiceView\_ViewPage
- Uncheck Generate Main

**Figure 51. Input Form window**

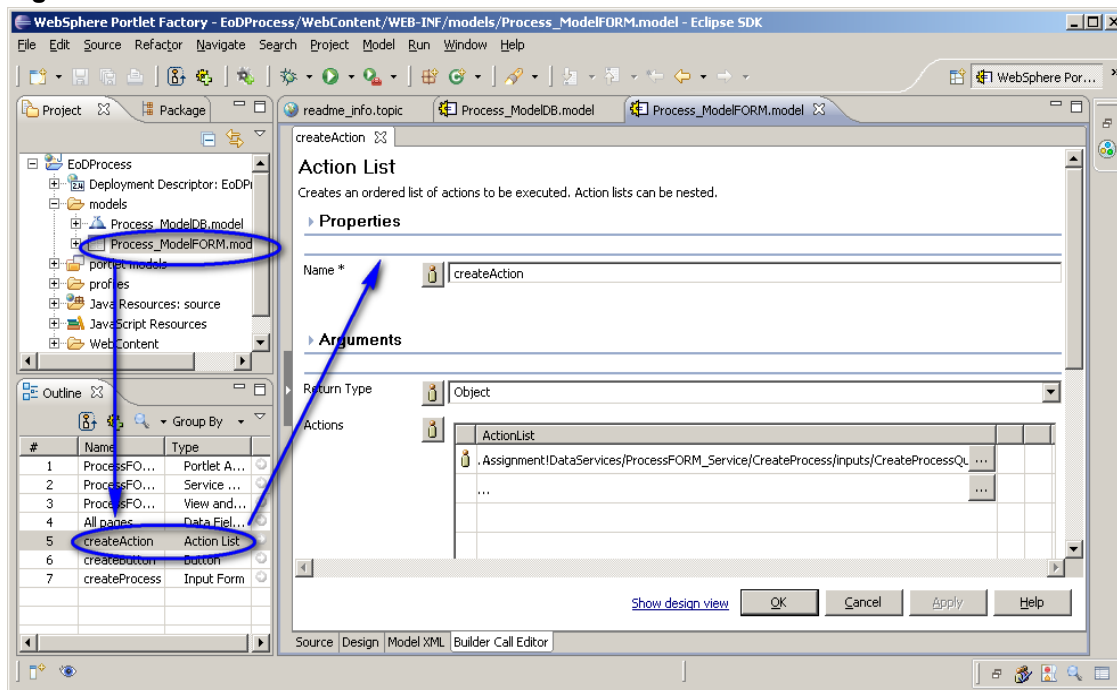




#### 1.4.4 Update Action List Builder

1. In the Project View, select Process\_ModelFORM and then, in the Outline view, double-click Builder Action List createAction (see figure 52).

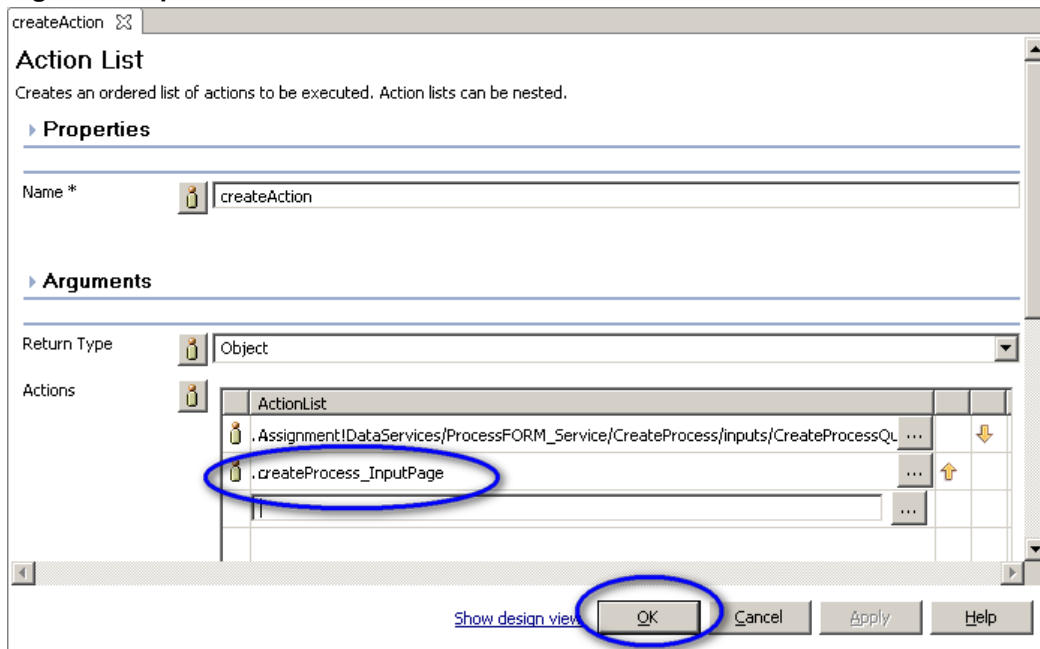
Figure 52. Show Action List window



2. Enter createProcess\_InputPage in the Actions values, and click OK (see figure 53).



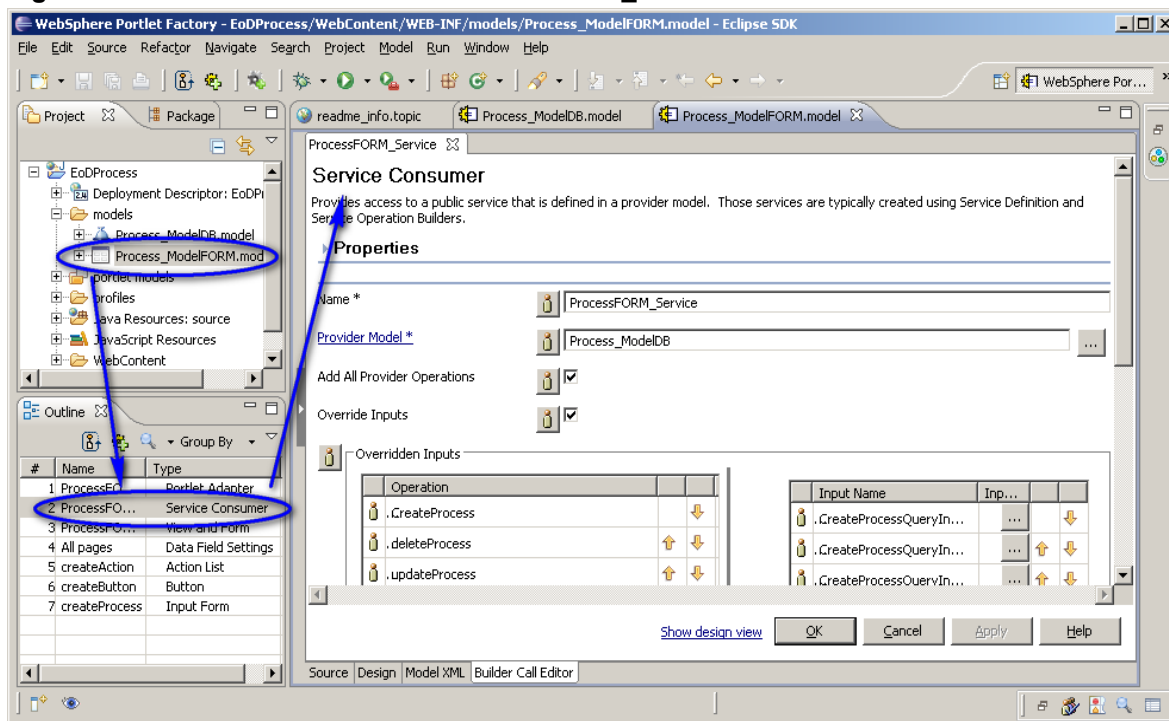
**Figure 53. Update Action List**



## 1.5 Update

1. Edit the Service Consumer ProcessFORM\_Service, double-clicking in the Process\_ModelFORM.model, and then double-clicking in the ProcessFORM\_Service Service Consumer (see figure 54).

**Figure 54. Edit Service Consumer ProcessFORM\_Service**



- In the Overridden Inputs section, update the parameters in Operation “updateProcess” with the following input values (see figure 55):

updateProcessQueryInputs.BEGINTIME:

`${DataServices/ProcessFORM_Service/getProcessDetail/results/Row/BeginTime}`

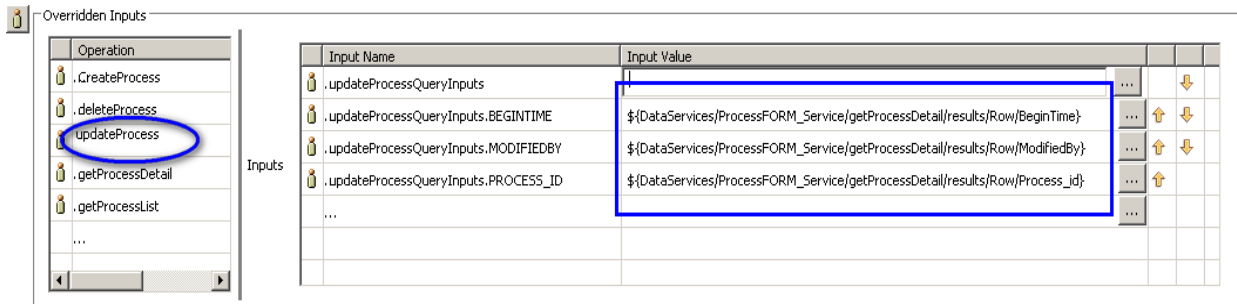
updateProcessQueryInputs. MODIFIEDBY:

`${DataServices/ProcessFORM_Service/getProcessDetail/results/Row/ModifiedBy}`

updateProcessQueryInputs.PROCESS\_ID:

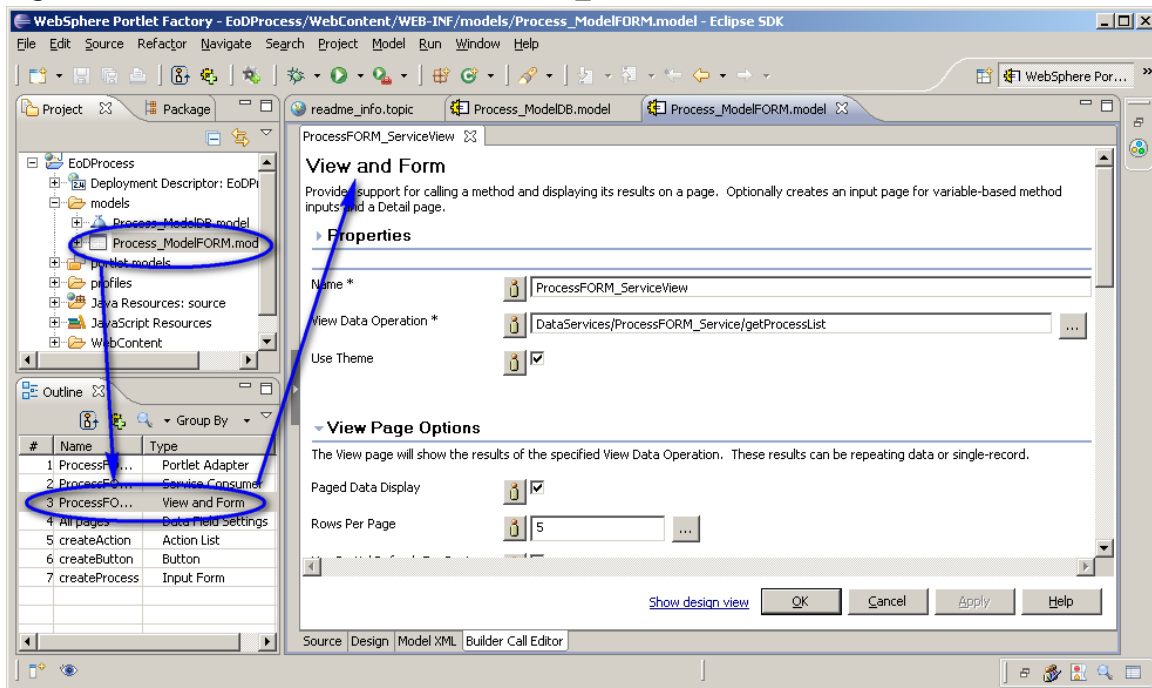
`${DataServices/ProcessFORM_Service/getProcessDetail/results/Row/Process_id}`

**Figure 55. Updating parameters**



- Edit the View and Form ProcessFORM\_ServiceView, double-clicking in the Process\_ModelFORM.model, and then double-clicking in the ProcessForm\_ServiceView View and Form (see figure 56).

**Figure 56. Edit View and Form ProcessFORM\_ServiceView**

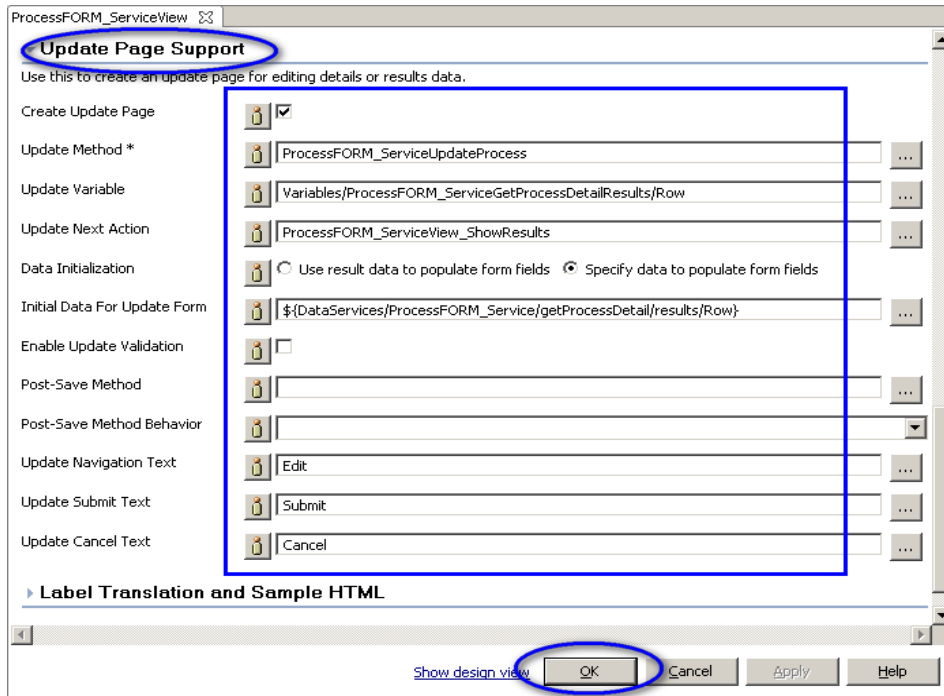


4. In the Update page Support section, enable the Create Update Page option and update these parameters with the following inputs (see figure 57):

- Update Method: ProcessFORM\_ServiceUpdateProcess
- Update Variable: Variables/ProcessFORM\_ServiceGetProcessDetailResults/Row
- Update Next Action: ProcessFORM\_ServiceView\_ShowsResults
- Data Initialization: Specify data to populate form fields
- Initial data For Update Form:

`#{DataServices/ProcessFORM_Service /getProcessDetail/results/Row}`

**Figure 57. Update Page Support section**

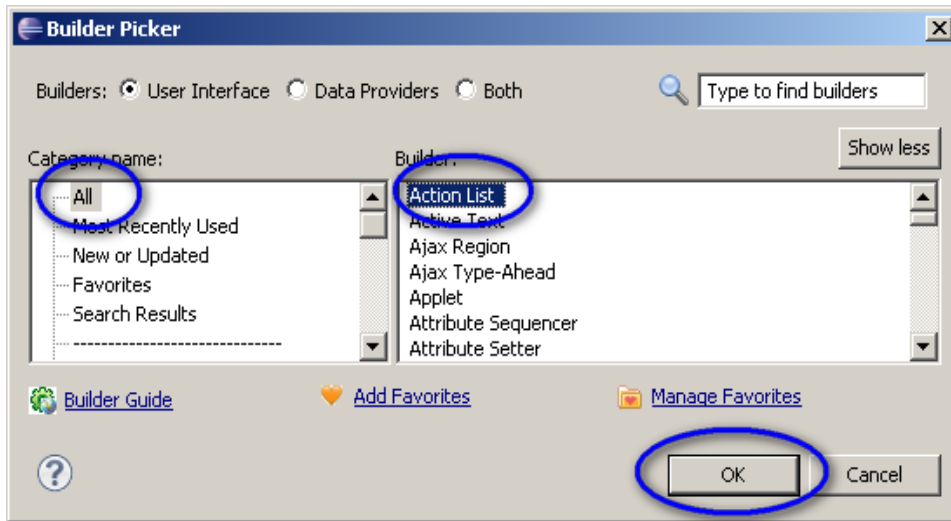


## 1.6 Delete

### 1.6.1 Add Action List Builder

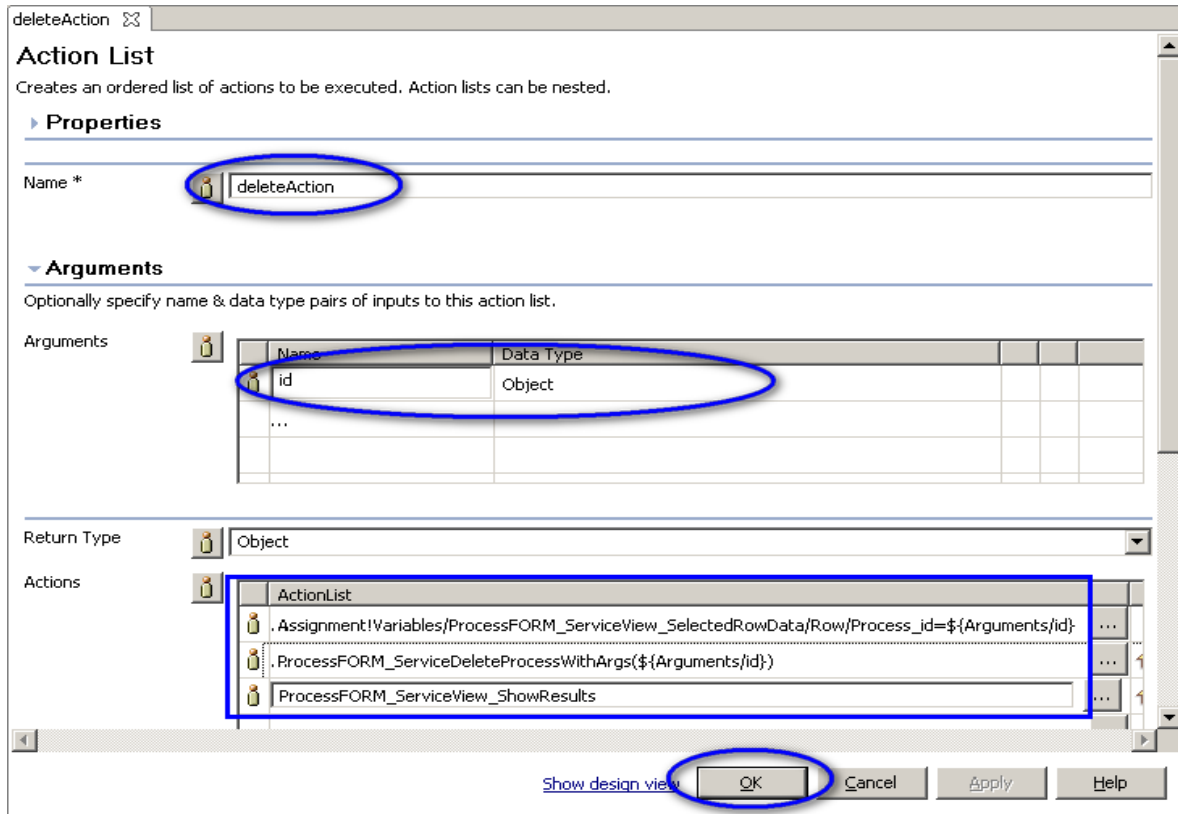
1. In the Outline View, click the Builder picker icon, select “Action List” from the Builder list, and click OK (see figure 58).

**Figure 58. Action List builder selected**



2. Enter "deleteAction" for the Action List Name, add "id" in the Arguments section, and add the three following actions (see figure 59):
  - Action 1: Assignment!  
Variables/ProcessFORM\_ServiceView\_SelectedRowData/Row/Process\_id=\${Arguments/id}
  - Action 2: ProcessFORM\_ServiceDeleteProcessWithArgs(\${Arguments/id})
  - Action 3: ProcessFORM\_ServiceView\_ShowResults

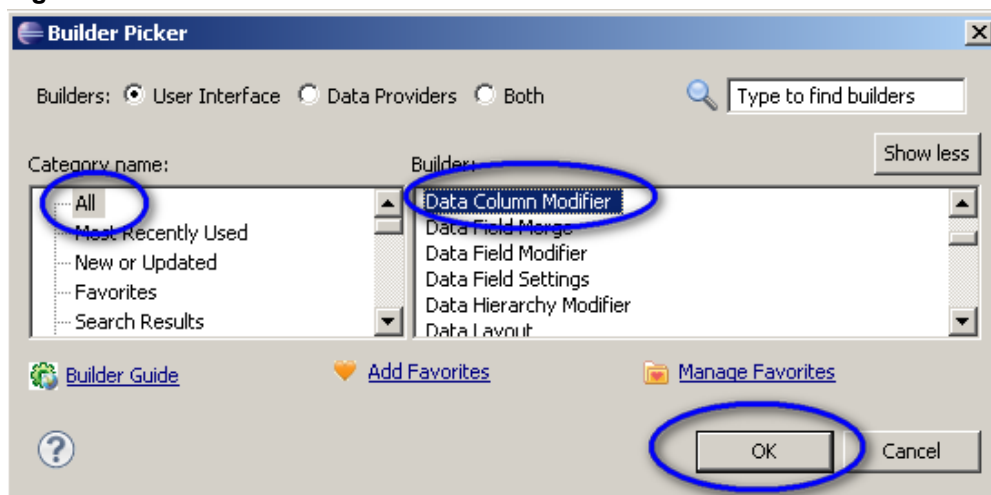
**Figure 59. Action List window**



### 1.1.2 Add Data Column Modifier Builder

1. In the Outline View, click the Builder picker icon, select Data Column Modifier from the Builder list, and click OK (see figure 60).

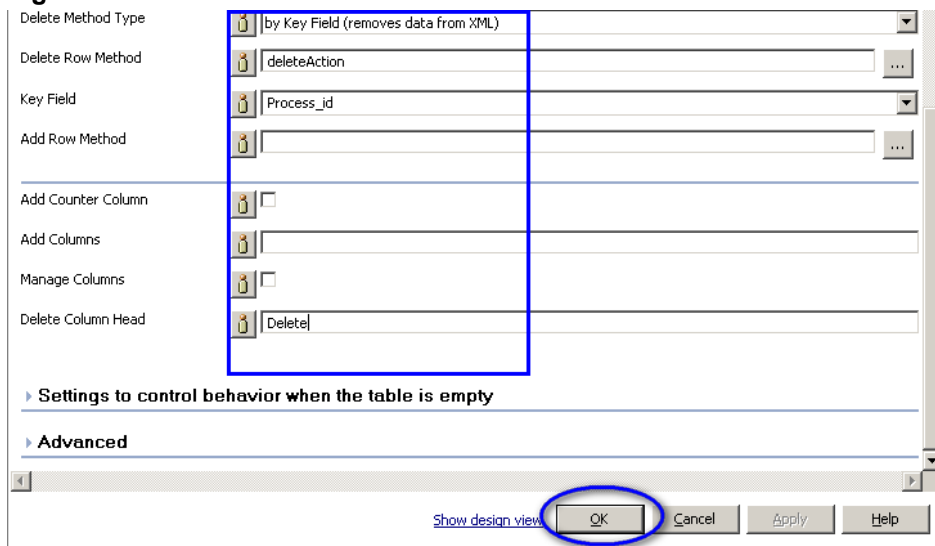
**Figure 60. Data Column Modifier selected**



2. Enter “deleteProcess” in the Name field, and add the following inputs for these parameters (see figure 61):

- Container Field:
- [ProcessFORM\_ServiceView\_ViewPage]ProcessFORM\_ServiceView\_ViewPage/RowSet/ROW
- Delete Column Style: Link
- Delete Link Text: Delete
- Delete Method Type: by Key Field (remove data from XML)
- Delete Row Method: deleteAction
- Key Field: Process\_id
- Delete Column Head: Delete

**Figure 61. Data Column Modifier window**



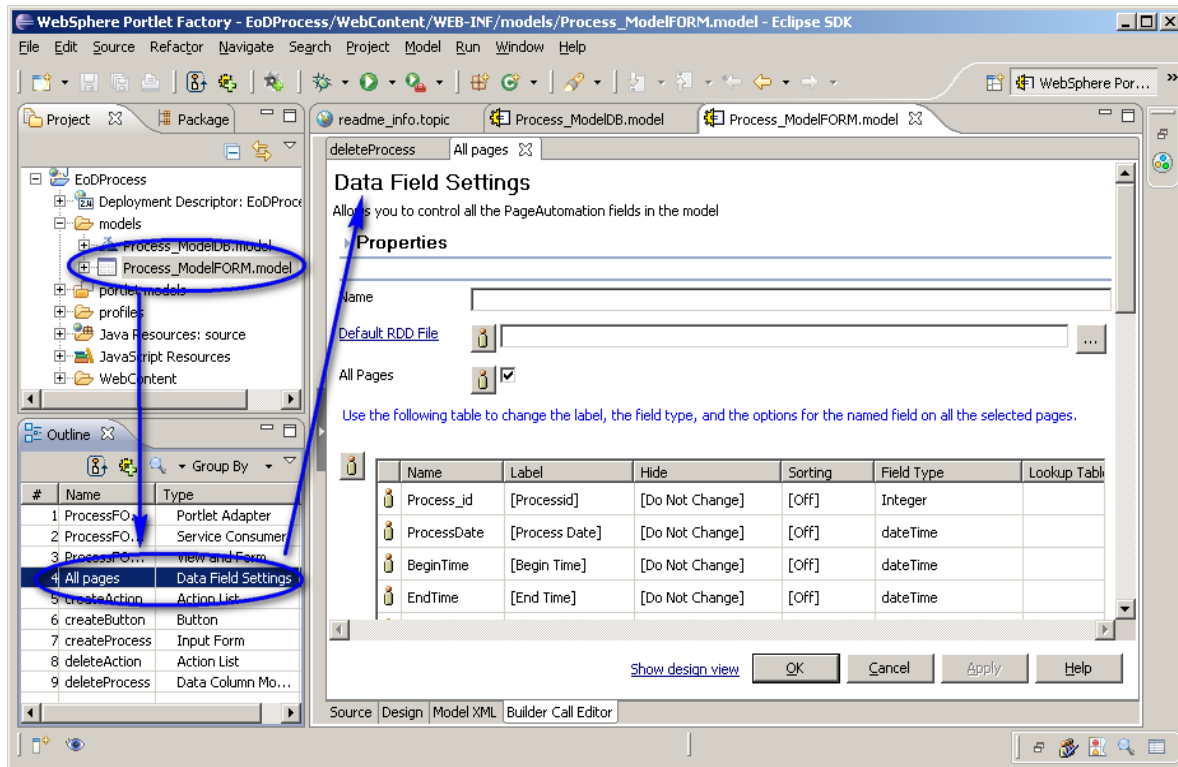
## 2 Filter using select input

### 2.1 Edit the Data Field Settings Builder

To do this:

1. Under the Project tab, double-click on Process\_ModelFORM.model, and then double-click on All pages / Data Field Settings (see figure 62).


**Figure 62. Edit Data Field Settings**



2. Enter Process\_Settings in the Name field, enable the All Pages option, and add the following inputs for these parameters (see figure 63):

- Year: Select (drop down); 2010,2011,2012,2013; required; RegularExpression; [0-9][0-9][0-9][0-9]
- Month: Select (drop down); Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec; required; RegularExpression, [A-Z][a-z][a-z]

**Figure 63. Data Field Settings window**


Process\_Settings 


### Data Field Settings

Allows you to control all the PageAutomation fields in the model
















▶ **Properties**

Name











Default RDD File   ...

All Pages  

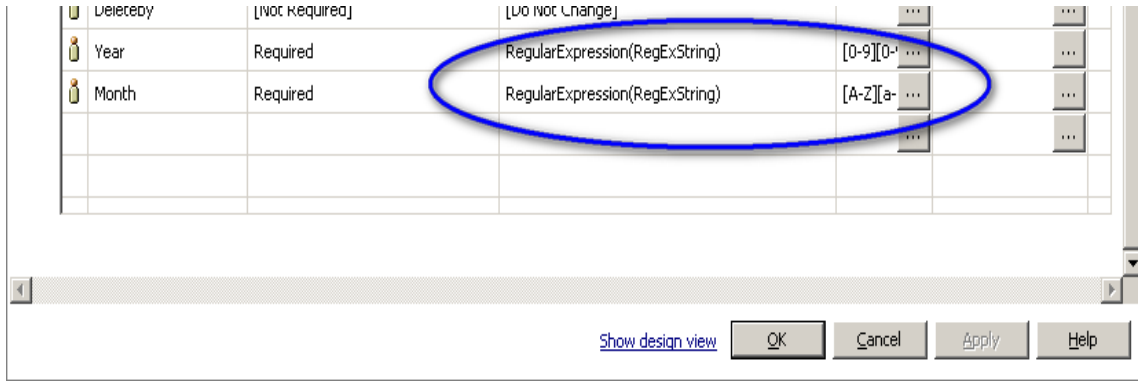
Use the following table to change the label, the field type, and the options for the named field on all the selected pages.

	Name	Label	Hide	Sorting	Field Type	Lookup Table / Choices
	Process_id	[Processid]	[Do Not Change]	[Off]	Integer	
	ProcessDate	[Process Date]	[Do Not Change]	[Off]	dateTime	
	BeginTime	[Begin Time]	[Do Not Change]	[Off]	dateTime	
	EndTime	[End Time]	[Do Not Change]	[Off]	dateTime	
	CreatedDate	[Created Date]	[Do Not Change]	[Off]	dateTime	
	CreatedBy	[Created By]	[Do Not Change]	[Off]	String	
	ModifiedDate	[Modified Date]	[Do Not Change]	[Off]	dateTime	
	ModifiedBy	[Modified By]	[Do Not Change]	[Off]	String	
	DeleteFlag	[Delete Flag]	[Do Not Change]	[Off]	Boolean (Checkbox)	
	Deleteby	[Deleteby]	[Do Not Change]	[Off]	String	
	Row_PageAut...	[Delete]	[Do Not Change]	[Off]	String	
	Year	[Year]	[Do Not Change]	[Off]	Select (drop down)	2010,2011,2012,20113
	Month	[Month]	[Do Not Change]	[Off]	Select (drop down)	Jan, Feb, Mar, Apr, May, J...
		[Blank]				

The following table applies only to Data Entry fields, and will affect those fields on all the selected pages. Note that choosing a Field Type above might already indicate a validation operation, so you only have to choose one here if you want something other than what the Field Type indicates.

	Name	Required	Validation Op	Argument	Error Message
	Process_id	[Not Required]	[Do Not Change]	...	...
	ProcessDate	[Not Required]	[Do Not Change]	...	...
	BeginTime	[Not Required]	[Do Not Change]	...	...
	EndTime	[Not Required]	[Do Not Change]	...	...
	CreatedDate	[Not Required]	[Do Not Change]	...	...
	CreatedBy	[Not Required]	[Do Not Change]	...	...
	ModifiedDate	[Not Required]	[Do Not Change]	...	...
	ModifiedBy	[Not Required]	[Do Not Change]	...	...
	DeleteFlag	[Not Required]	[Do Not Change]	...	...

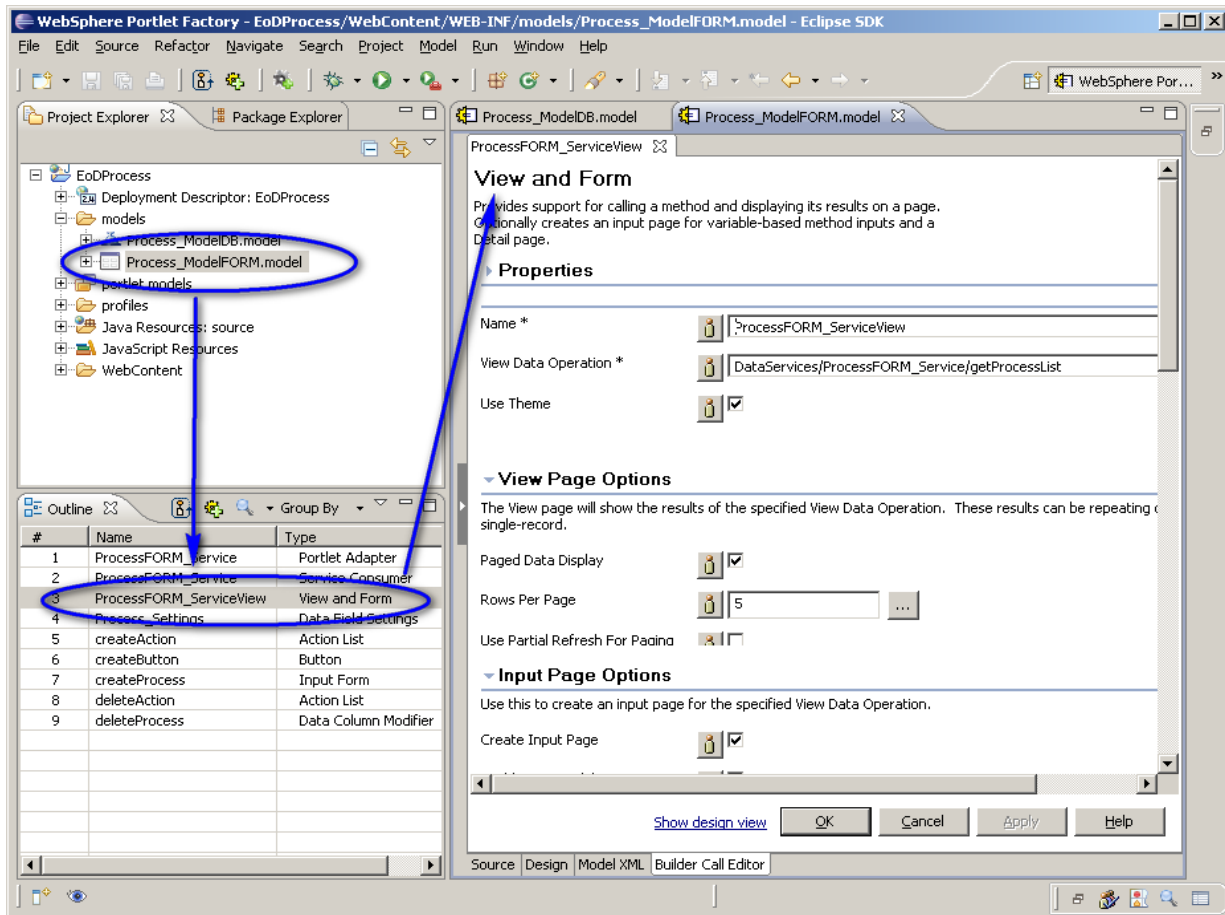




## 2.2 Edit View and Form ProcessFORM\_ServiceView

- Under the Project Explorer tab, double-click on `Process_ModelFORM.model`, and then double-click on `ProcessFORM_ServiceView / View and Form` (see figure 64).

Figure 64. Edit View and Form



- In the Input Page Option section, update with Input Submit Text and Back Button Text fields as shown in figure 65.

**Figure 65. Input Page Option section**

**Input Page Options**

Use this to create an input page for the specified View Data Operation.

Create Input Page

Enable Input Validation

Post-Save Method

Post-Save Method Behavior

Input Submit Text

Back Button Text

Insert Input Form In View Page

3. To run the model, click the green arrow button, enter the Year and Month, and then click the Search button (see figure 66).

**Figure 66. Filter window**

http://localhost:10039/EoDProcess/webengine/Process\_ModelFORM - Windows Internet Explorer

http://localhost:10039/EoDProcess/webengine/Pro

Year \* 2012

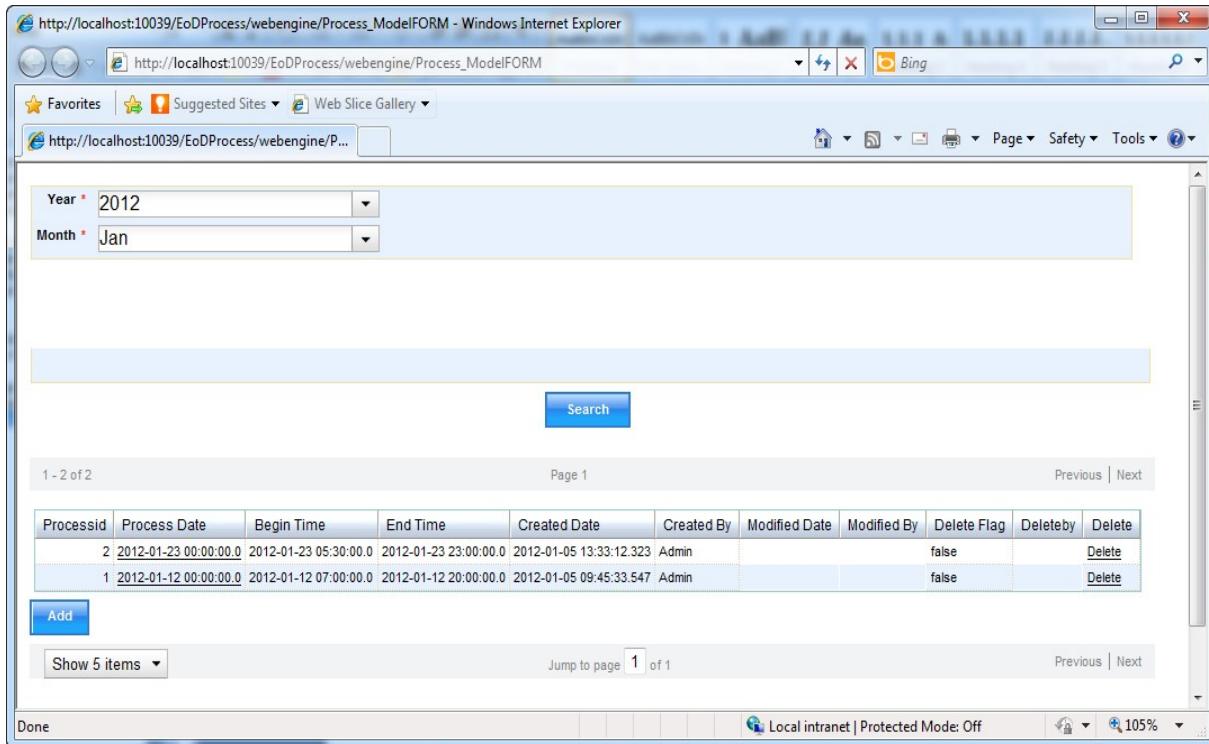
Month \* Jan

Search

Done Local intranet | Protected Mode: Off 105%

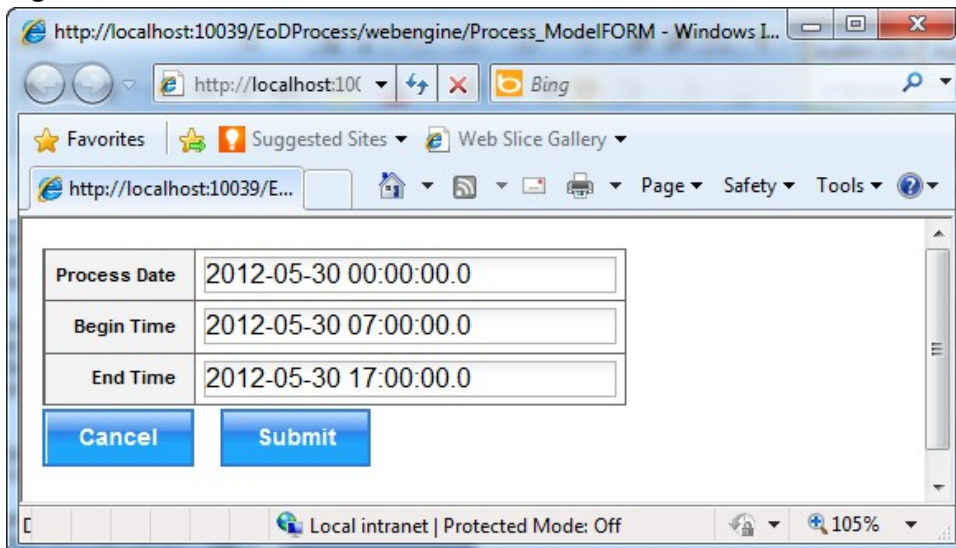
4. In the next window, click the Add button, as shown in figure 67.

**Figure 67. List window**



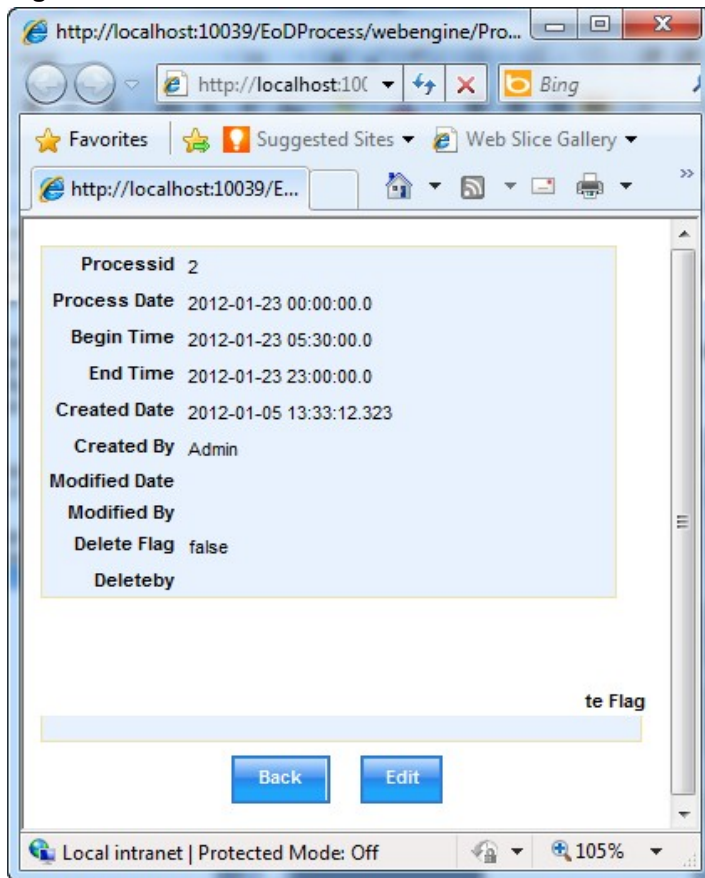
5. Enter the values for Process Date, Begin Time, and End Time as shown in figure 68. Click the Submit button to save this data, or click the Cancel button to cancel.

**Figure 68. Add window**



6. Click Delete to delete the data, or click 2012-01-23 00:00:00.0 to view the Details window (see figure 69).

**Figure 69. Details window**



7. Click the Edit button to edit this data (see figure 70), or click the Back button to return to the list. Enter the value to edit and click Save, to save this data, or click Cancel, to cancel.

**Figure 70. Edit window**

The screenshot shows a web browser window with the following form fields and values:

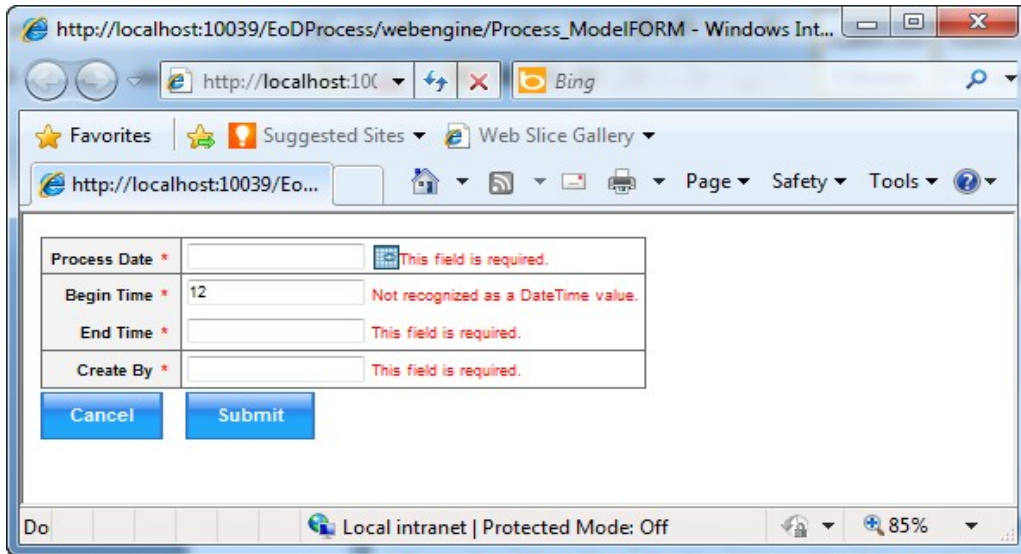
Processid	2
Process Date	2012-01-23 00:00:00.0
Begin Time	2012-01-23 05:30:00.0
End Time	2012-01-23 23:00:00.0
Created Date	2012-01-05 13:33:12.323
Created By	Admin
Modified Date	
Modified By	
Delete Flag	<input type="checkbox"/>
Deleteby	

Buttons: Cancel, Submit

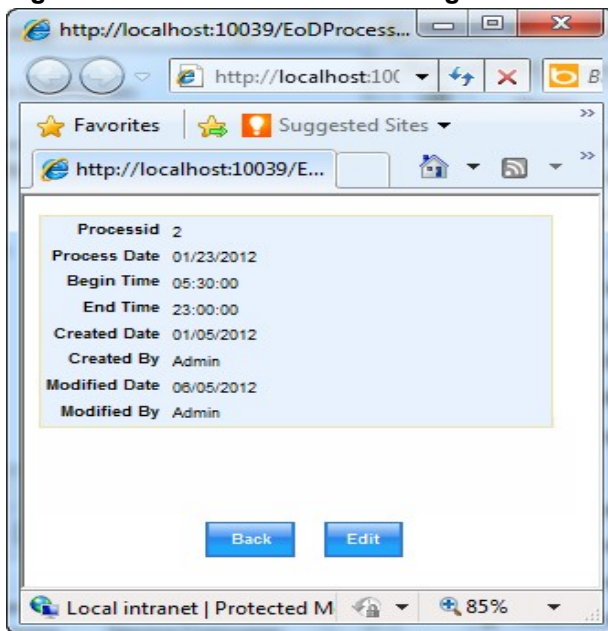
### **3 Add field validation in the portlet input form**

In this section we explain the steps to add field validation in the portlet input form in WEF (see figures 71 and 72).

**Figure 71. Simple data validation for required and DateTime data type**



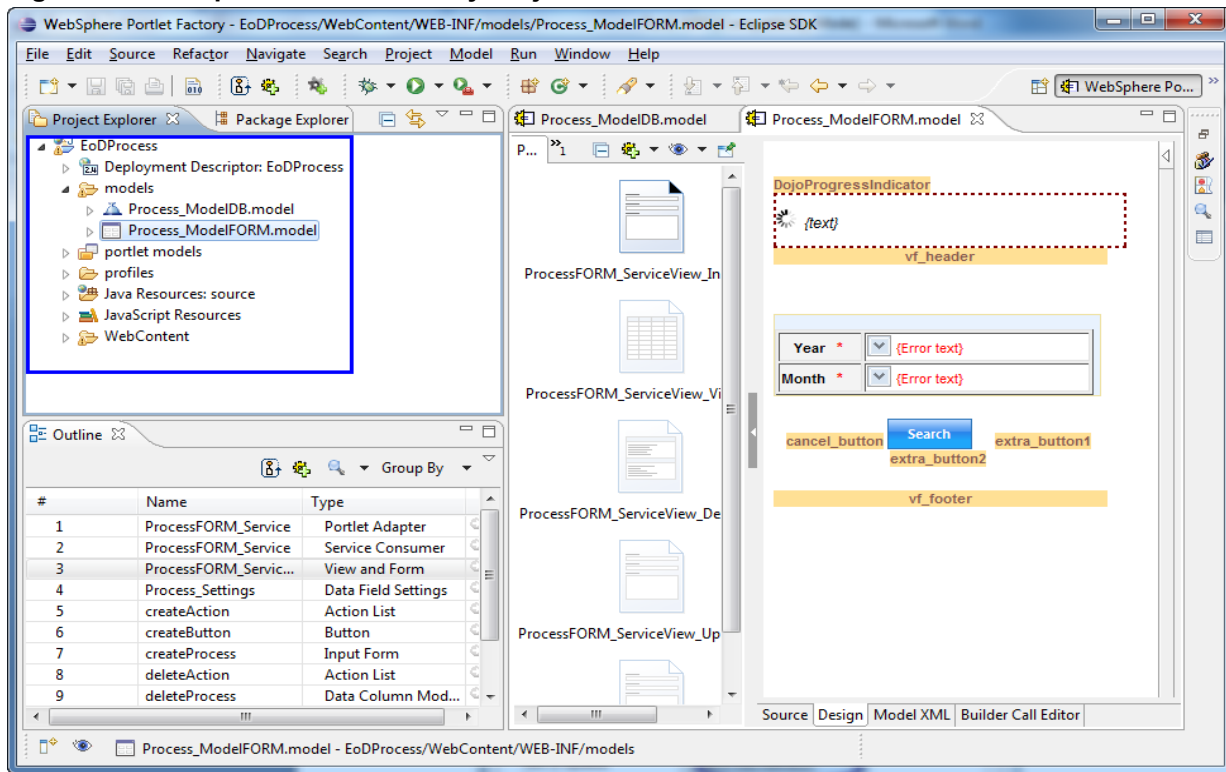
**Figure 72. DateTime data formatting**



To begin the process of adding field validation to the portlet input form, open the Portlet Factory Project "EoDProcess" that we created in Section 1 (see figure 73).



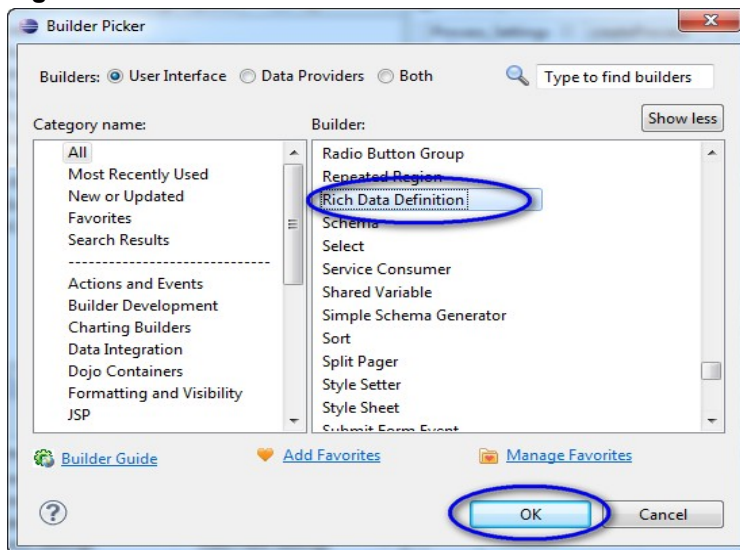
**Figure 73. WebSphere Portlet Factory Project “EoDProcess”**



### 3.1 Add the Rich Data Definition Builder for input form

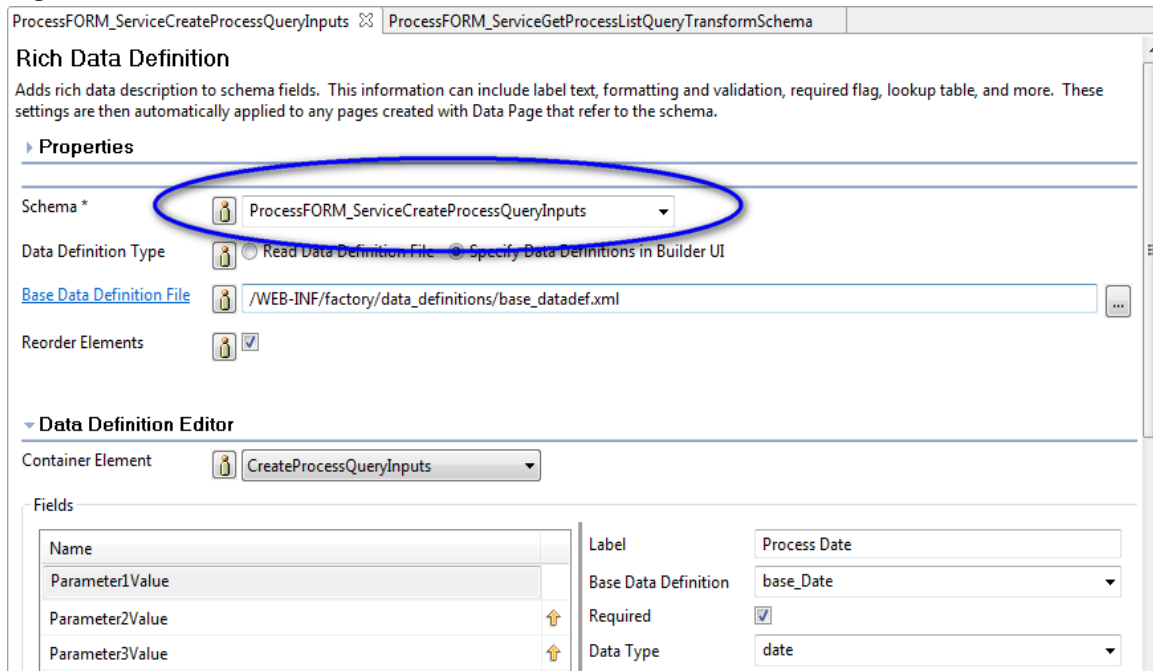
1. In the Outline View, click the builder picker icon and select “Rich Data Definition” from the Builder list; click OK (see figure 74).

**Figure 74. Rich Data Definition Builder selected**



2. Select ProcessFORM\_ServiceCreateProcessQueryInputs for the Schema (see figure 75).

**Figure 75. Rich Data Definition window**



3. In the Data Definition Editor section, select ProcessDate, updating the parameters with the following input values (see figure 76):

- Label: Process Date
- Base Data Definition: base\_date
- Enable "Required"
- Data Type: date
- Formatting Expression: Format(yyy-MM-dd\$MM/dd/yyyy)
- Validate Expression: Required Date(yyyy-MM-dd)



**Figure 76. Data Definition Editor section for ProcessDate**

▼ Data Definition Editor

Container Element CreateProcessQueryInputs

Fields

Name	
Parameter1Value	
Parameter2Value	↑
Parameter3Value	↑
Parameter4Value	↑

Label: Process Date

Base Data Definition: base\_Date

Required:

Data Type: date

Enumeration Values:

Hidden:

Read Only:

Formatting Expression: Format(yyyy-MM-dd\$Mm/dd/yyyy)

Validate Expression: Required Date(yyyy-MM-dd)

Translate Expression:

4. In the Data Definition Editor section, select BeginTime, updating the parameters with the following input values (see figure 77):

- Label: Begin Time
- Enable “Required”
- Data Type: date
- Formatting Expression: Time Only
- Validate Expression: Required Time Only

**Figure 77. Data Definition Editor section for BeginTime**

▼ Data Definition Editor

Container Element CreateProcessQueryInputs

Fields

Name	
Parameter1Value	
Parameter2Value	↑
Parameter3Value	↑
Parameter4Value	↑

Label: Begin Time

Base Data Definition:

Required:

Data Type: date

Enumeration Values:

Hidden:

Read Only:

Formatting Expression: Time Only

Validate Expression: Required Time Only

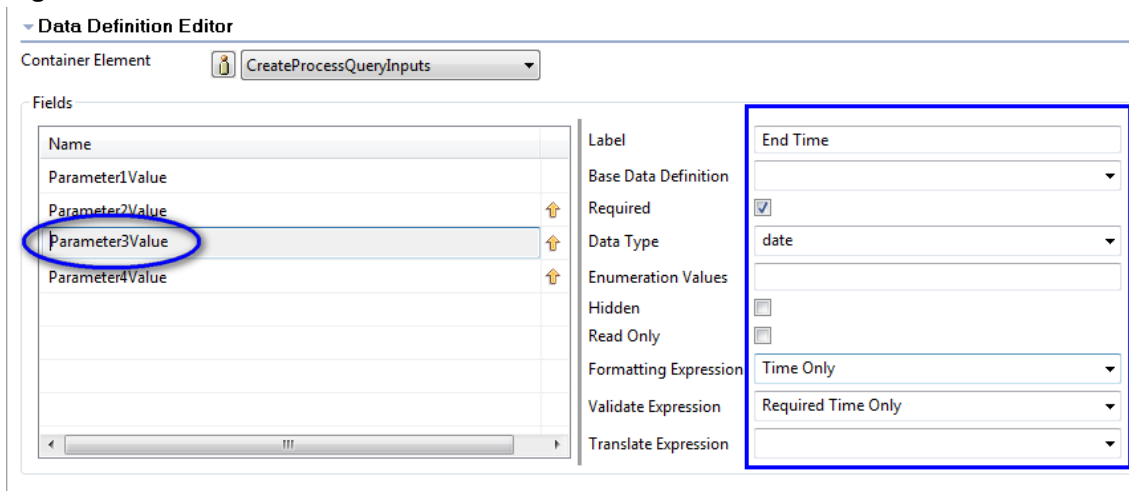
Translate Expression:

5. In the Data Definition Editor section, select EndTime, updating the parameters with the following input values (see figure 78):

- Label: End Time
- Enable “Required”
- Data Type: date

- Formatting Expression: Time Only
- Validate Expression: Required Time Only

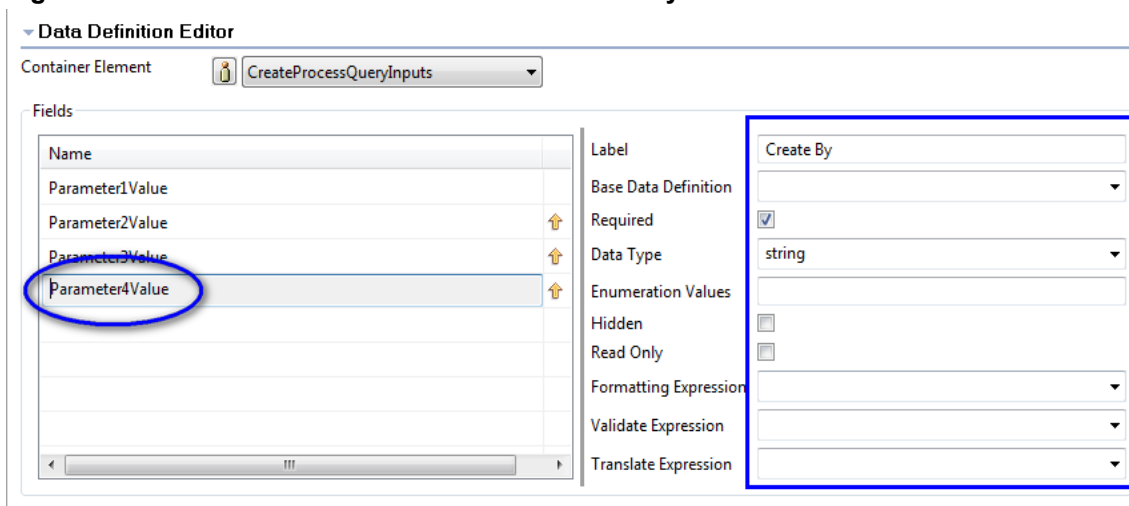
**Figure 78. Data Definition Editor section for EndTime**



6. Still in the Data Definition Editor section, select CreateBy, updating the parameters with the following input values (see figure 79):

- Label: Create By
- Enable “Required”
- Data Type: string

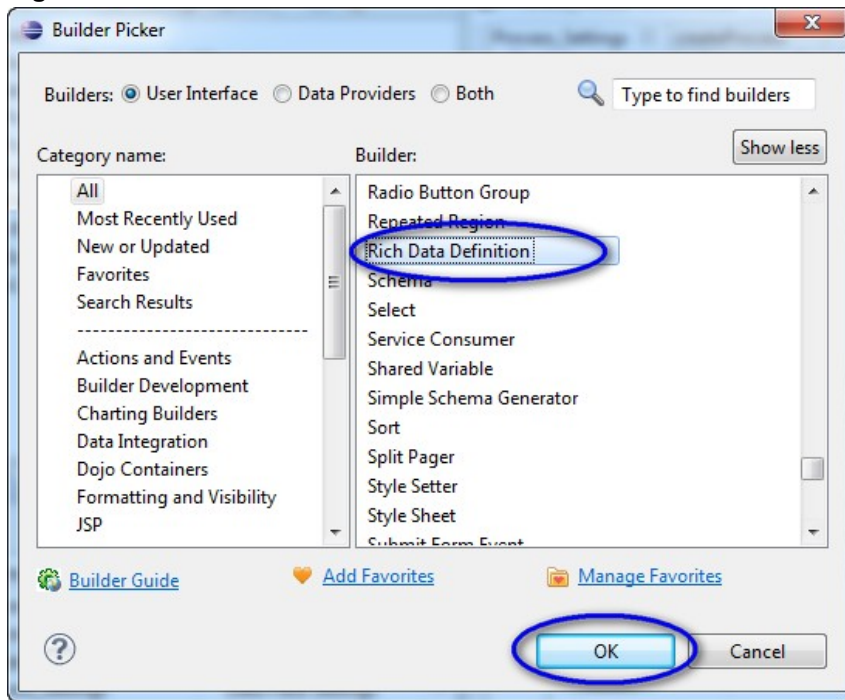
**Figure 79. Data Definition Editor section for CreateBy**



### **3.2 Add the Rich Data Definition Builder for list form**

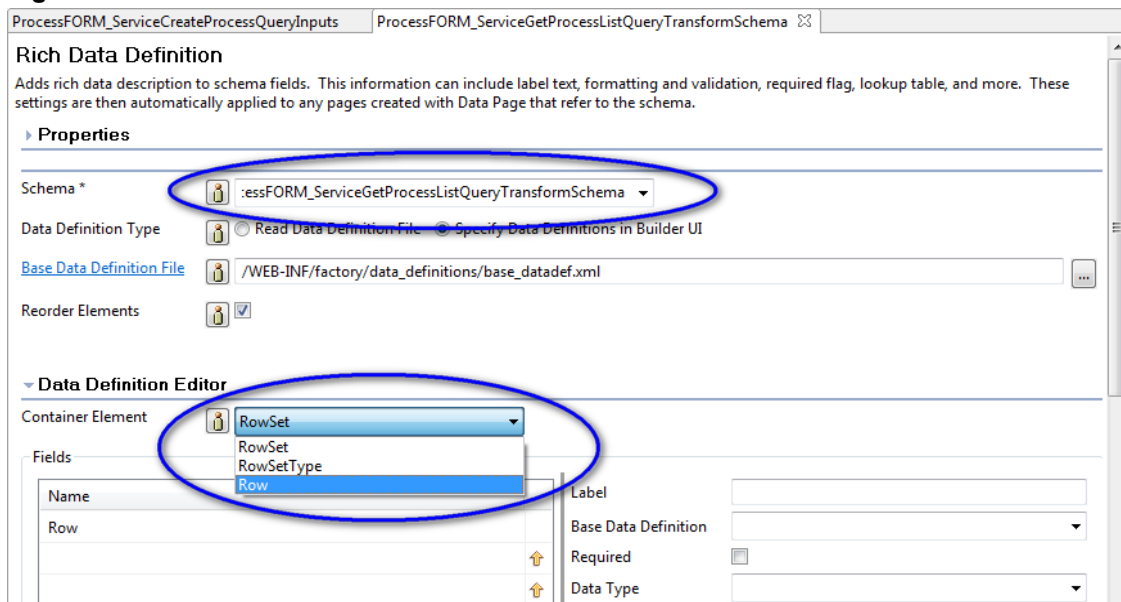
1. In the Outline View, click the Builder picker icon and select “Rich Data Definition” from the Builder list; click OK (see figure 80).

**Figure 80. Rich Data Definition Builder selected**



2. Select ProcessFORM\_ServiceGetProcessListQueryTransformSchema for the Schema, and select Row for the Container Element (see figure 81).

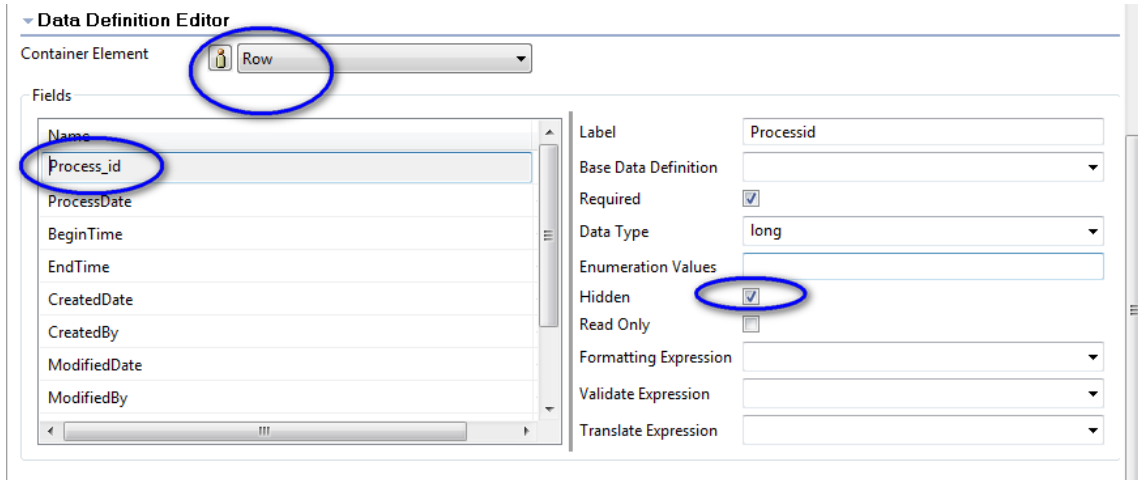
**Figure 81. Rich Data Definition window**



3. In the Data Definition Editor section, select Process\_id, updating the parameters with the following input values (see figure 82):

- Label: Processid
- Enable “Required” and “Hidden”
- Data Type: long

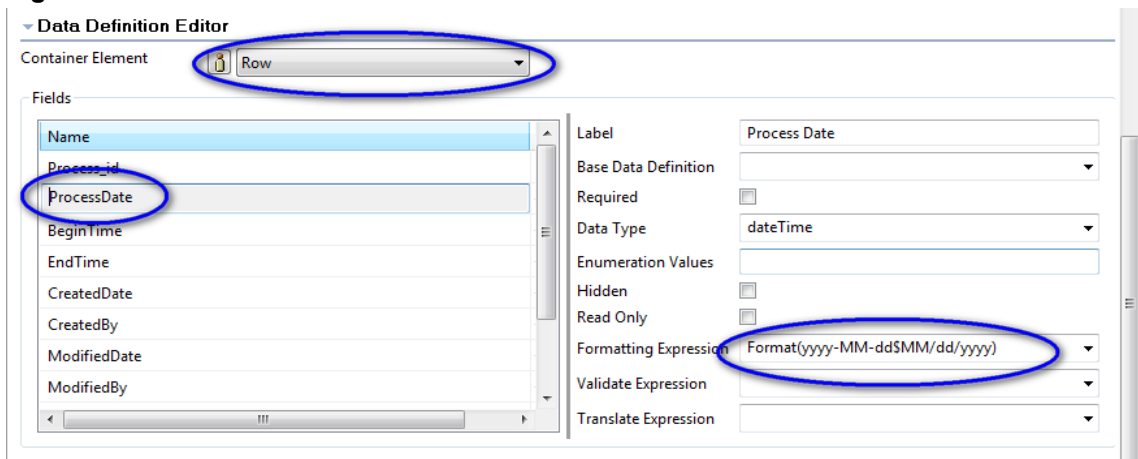
**Figure 12. Data Definition Editor section for Process\_id**



4. In the Data Definition Editor section, select ProcessDate, updating the parameters with the following input values (see figure 83).

- Label: Process Date
- Data Type: dateTime
- Formatting Expression: Format(yyyy-MM-dd\$MM/dd/yyyy)

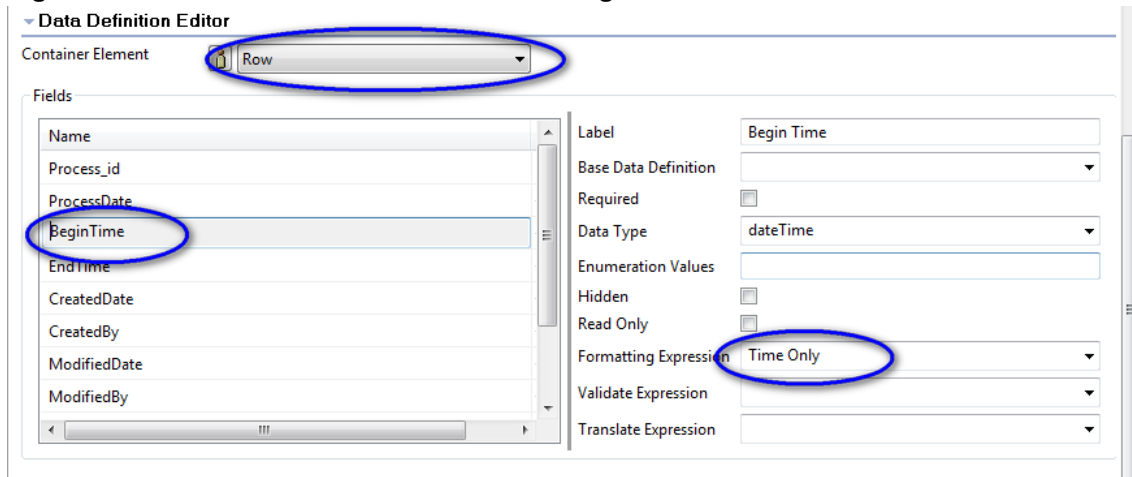
**Figure 83. Data Definition Editor section for ProcessDate**



5. In the Data Definition Editor section, select BeginTime, updating the parameters with the following input values (see figure 84):

- Label: Begin Time
- Data Type: dateTime
- Formatting Expression: Time Only

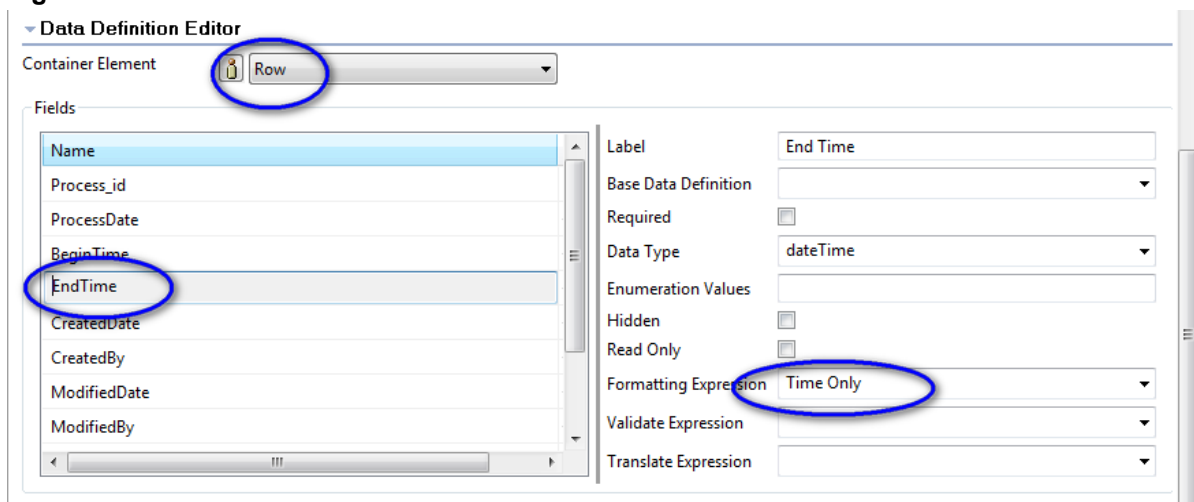
**Figure 84. Data Definition Editor section for BeginTime**



6. In the Data Definition Editor section, select EndTime, updating the parameters with the following input values (see figure 85):

- Label: End Time
- Data Type: dateTime
- Formatting Expression: Time Only

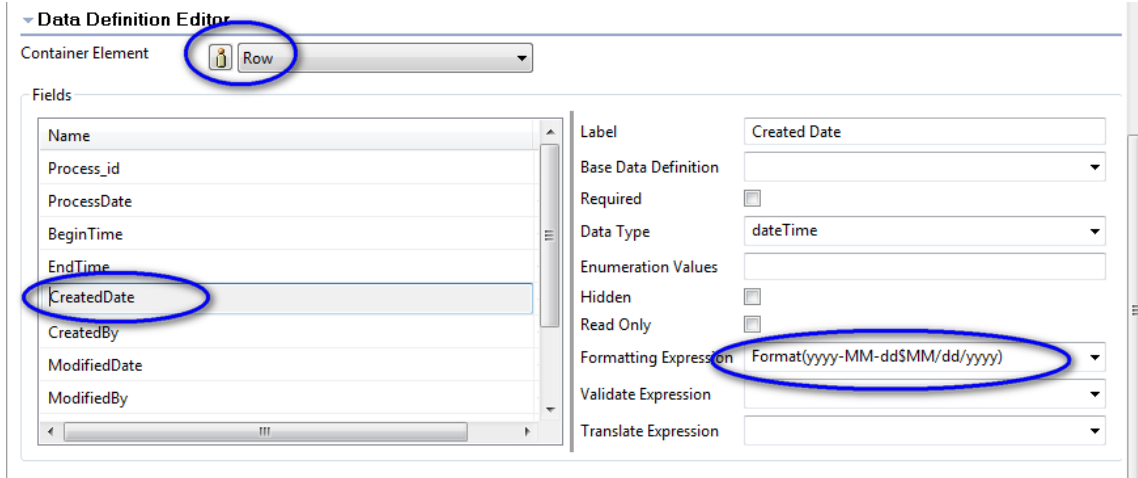
**Figure 85. Data Definition Editor section for EndTime**



7. In the Data Definition Editor section, select CreatedDate, updating the parameters with the following input values (see figure 86):

- Label: Created Date
- Data Type: dateTime
- Formatting Expression: Format(yyyy-MM-dd\$MM/dd/yyyy)

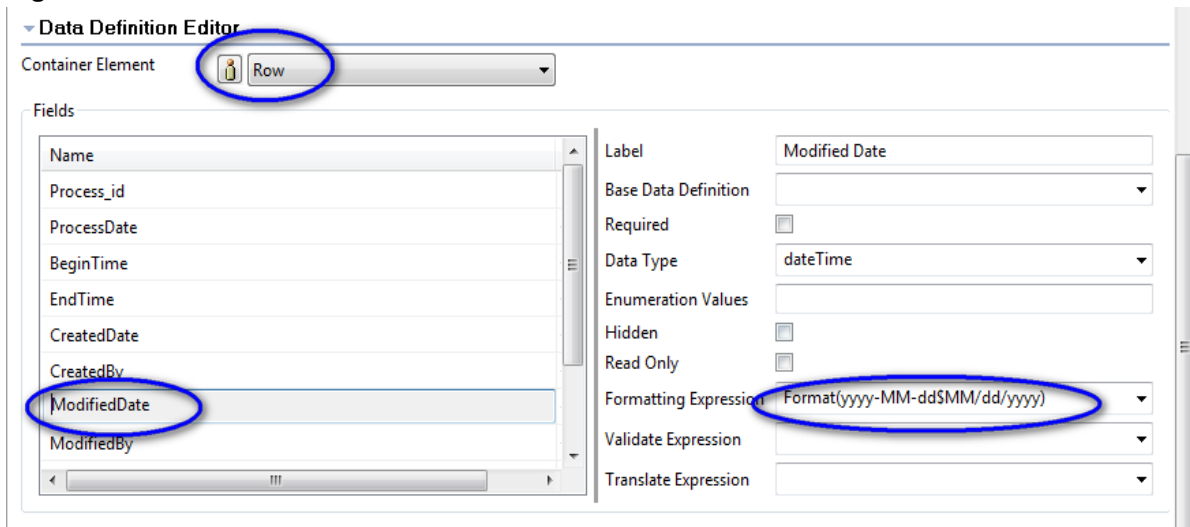
**Figure 86. Data Definition Editor section for CreatedDate**



8. In the Data Definition Editor section, select ModifiedDate, updating the parameters with the following input values (see figure 87):

- Label : Modified Date
- Data Type : dateTime
- Formatting Expression : Format(yyyy-MM-dd\$MM/dd/yyyy)

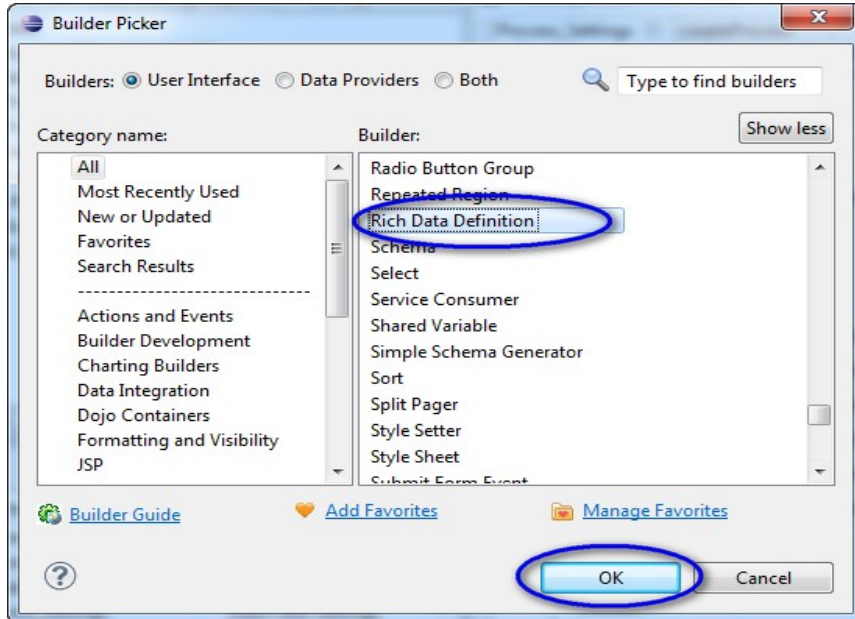
**Figure 87. Data Definition Editor section for ModifiedDate**



### 3.3 Add the Rich Data Definition Builder for edit form

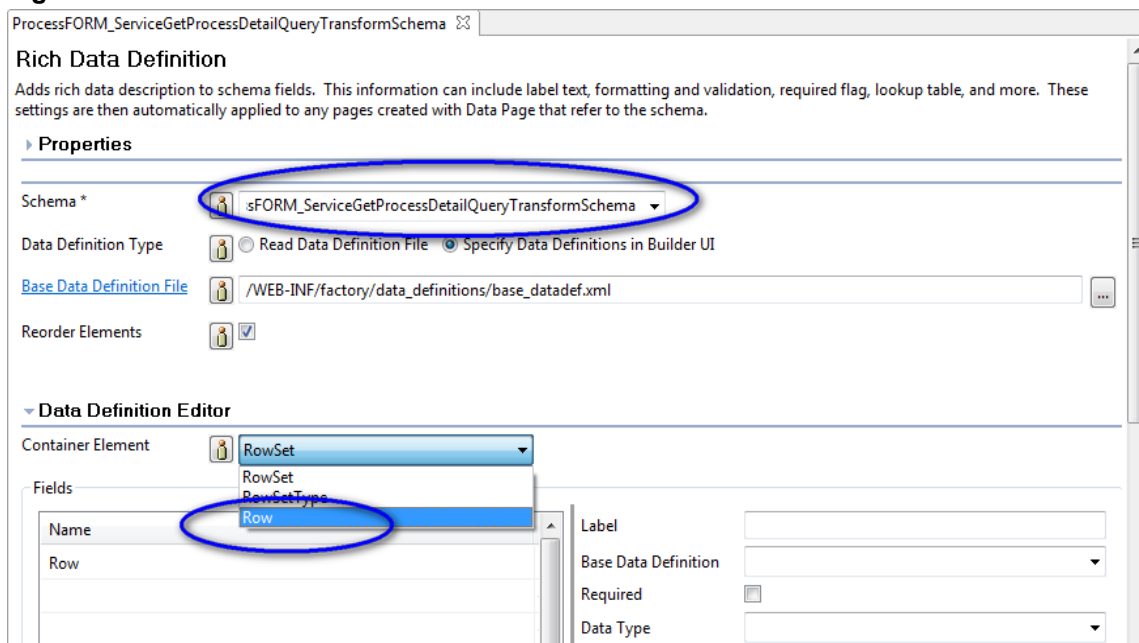
1. In the Outline View, click the Builder picker icon and select “Rich Data Definition” from the Builder list; click OK (see figure 88).

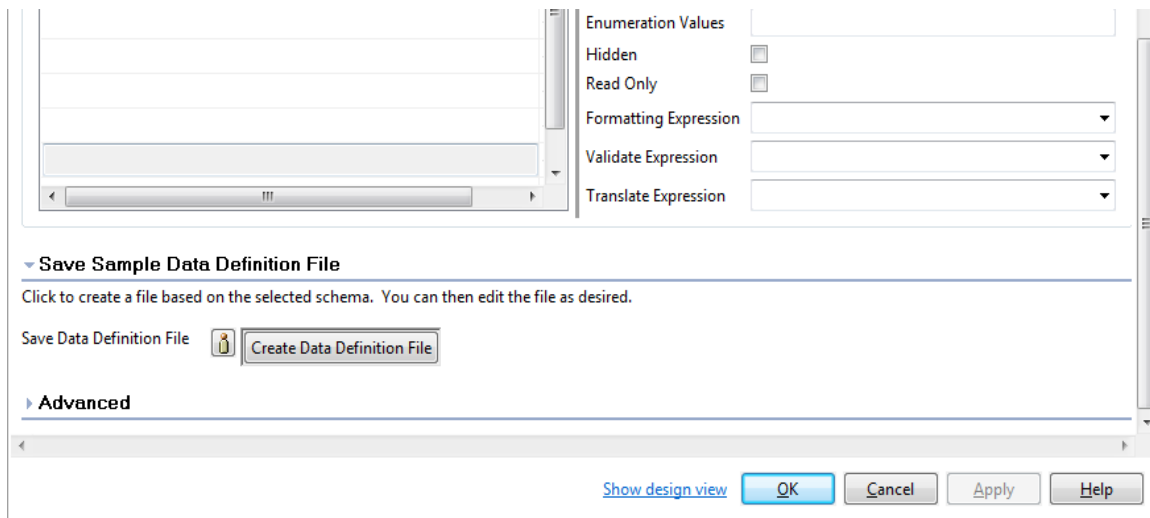
Figure 88. Rich Data Definition Builder selected



2. Select ProcessFORM\_ServiceGetProcessDetailQueryTransformSchema for the Schema, and select Row for the Container Element (see figure 89).

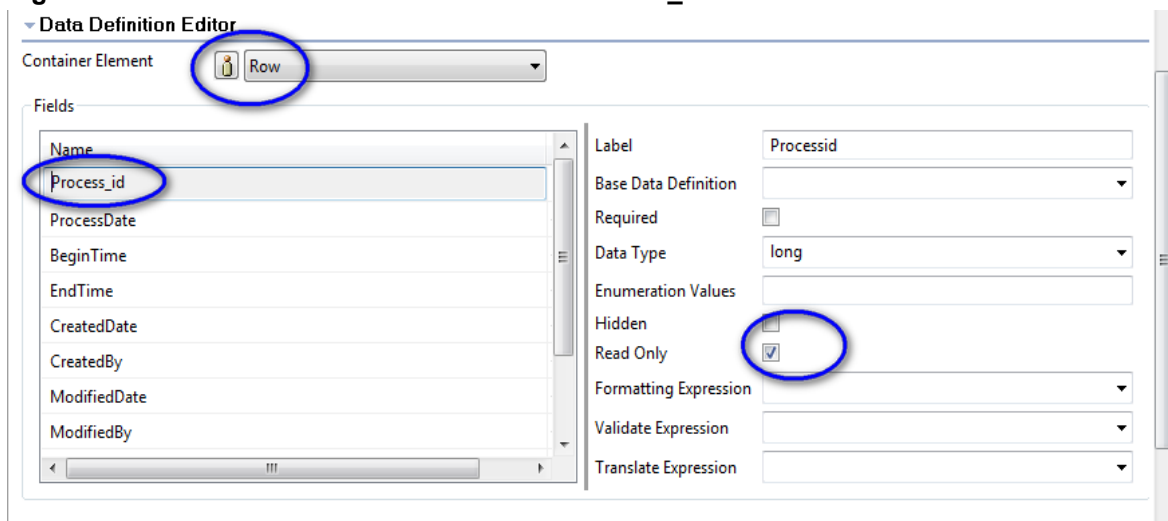
Figure 89. Rich Data Definition window





3. In the Data Definition Editor section, select Process\_id, updating the parameters with the following input values (see figure 90):
  - Label: Processid
  - Data Type: long
  - Enabled "Read Only"

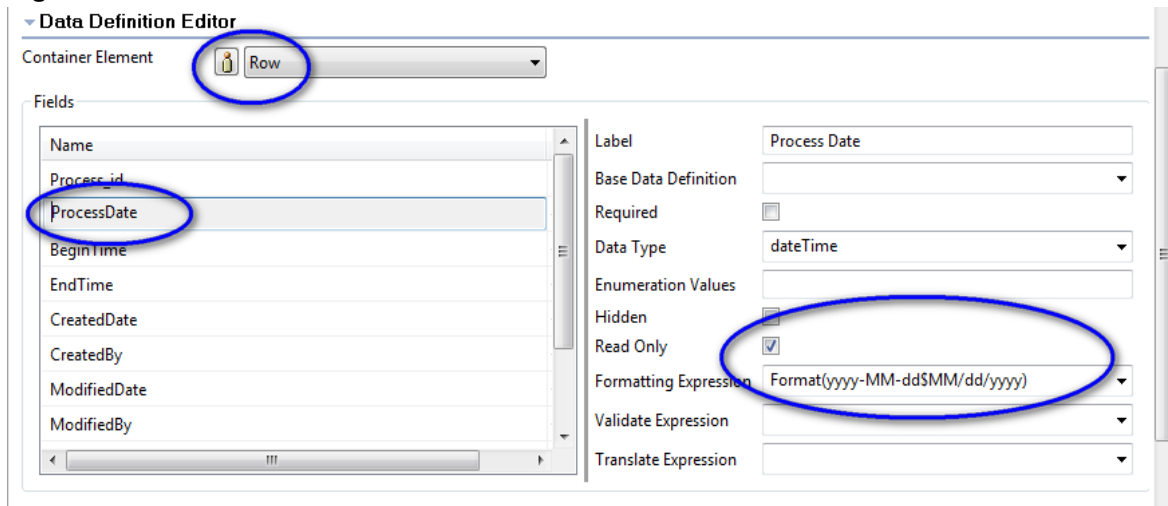
**Figure 90. Data Definition Editor section for Process\_id**



4. In the Data Definition Editor section, select ProcessDate, updating the parameters with the following inputs value (see figure 91):
  - Label: Process Date
  - Data Type: dateTime
  - Enabled "Read Only"
  - Formatting Expression: Format(yyyy-MM-dd\$MM/dd/yyyy)



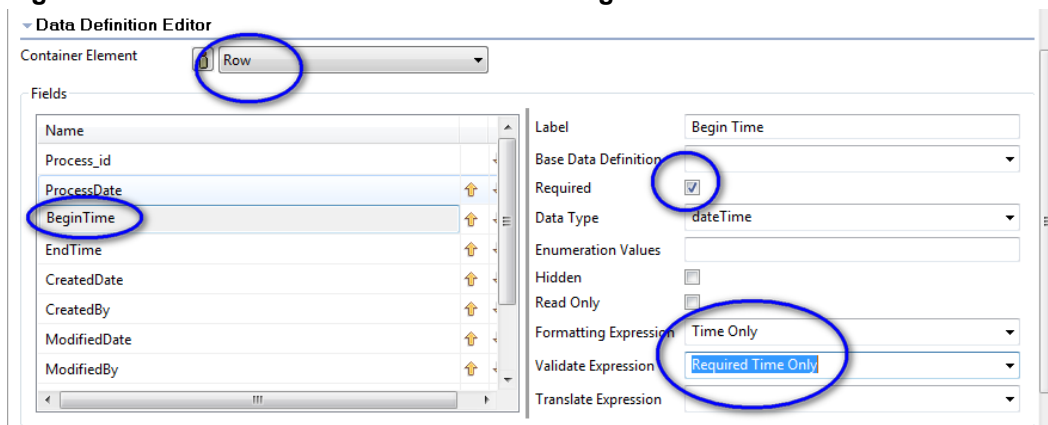
**Figure 91. Data Definition Editor for ProcessDate**



5. In the Data Definition Editor section, select BeginTime, updating the parameters with the following input values (see figure 92):

- Label: Begin Time
- Data Type: dateTime
- Enabled “Required”
- Formatting Expression: Time Only
- Validate Expression: Required Time Only

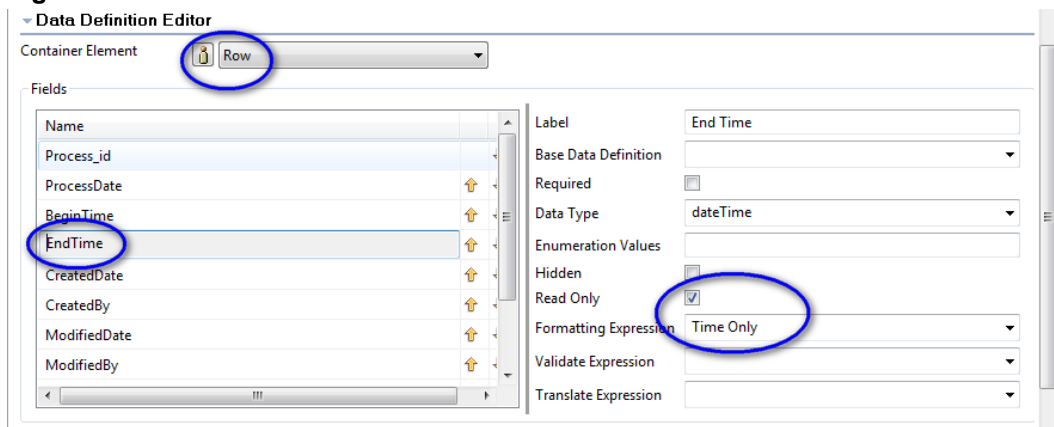
**Figure 92. Data Definition Editor section for BeginTime**



6. In the Data Definition Editor section, select EndTime, updating the parameters with the following input values (see figure 93):

- Label: End Time
- Enabled “Read Only”
- Data Type: dateTime
- Formatting Expression: Time Only

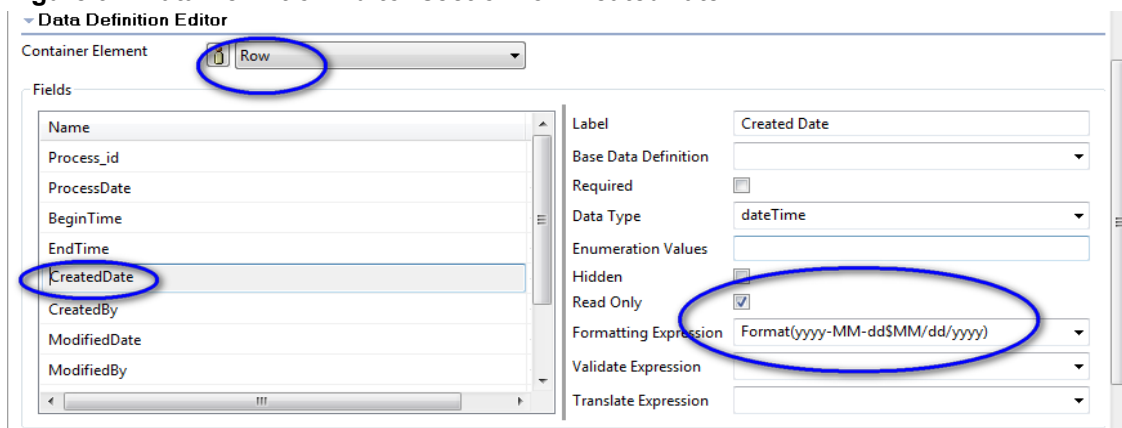
**Figure 93. Data Definition Editor section for EndTime**



7. In the Data Definition Editor section, select CreatedDate, updating the parameters with the following input values (see figure 94):

- Label: Created Date
- Data Type: dateTime
- Enabled "Read Only"
- Formatting Expression: Format(yyyy-MM-dd\$MM/dd/yyyy)

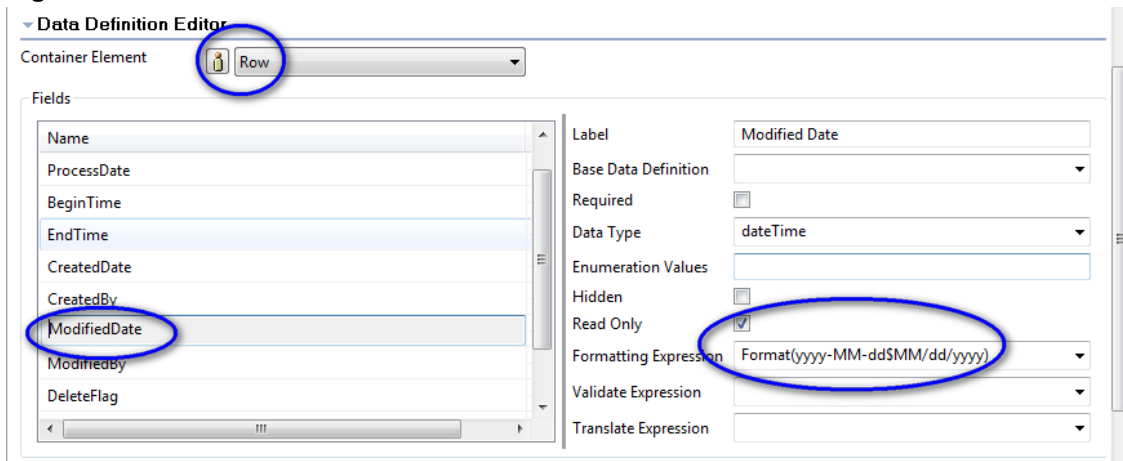
**Figure 94. Data Definition Editor section for CreatedDate**



8. In the Data Definition Editor section, select ModifiedDate, updating the parameters with the following input values (see figure 95):

- Label: Modified Date
- Data Type: dateTime
- Enabled "Read Only"
- Formatting Expression: Format(yyyy-MM-dd\$MM/dd/yyyy)

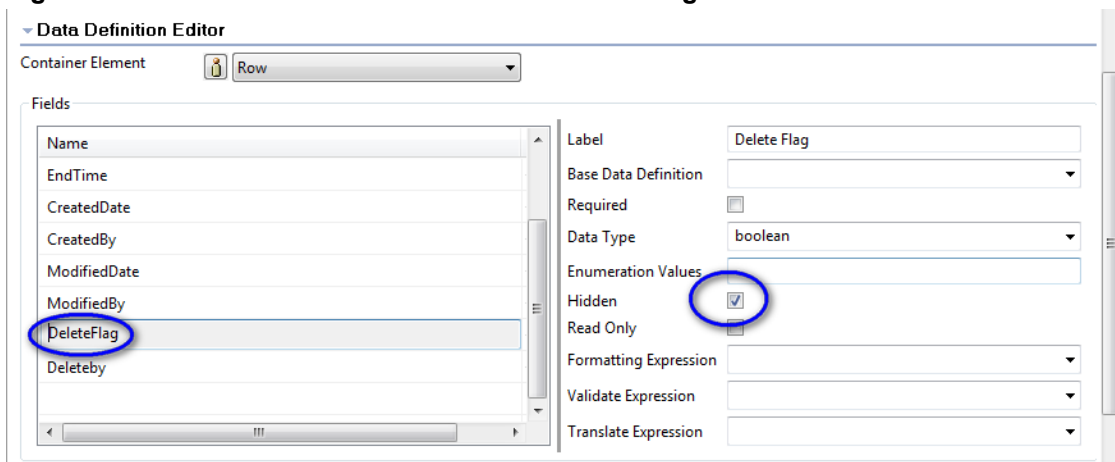
**Figure 95. Data Definition Editor section for ModifiedDate**



9. In the Data Definition Editor section, select DeleteFlag, updating the parameters with the following input values (see figure 96):

- Label: Delete Flag
- Data Type: boolean
- Enabled "Hidden"

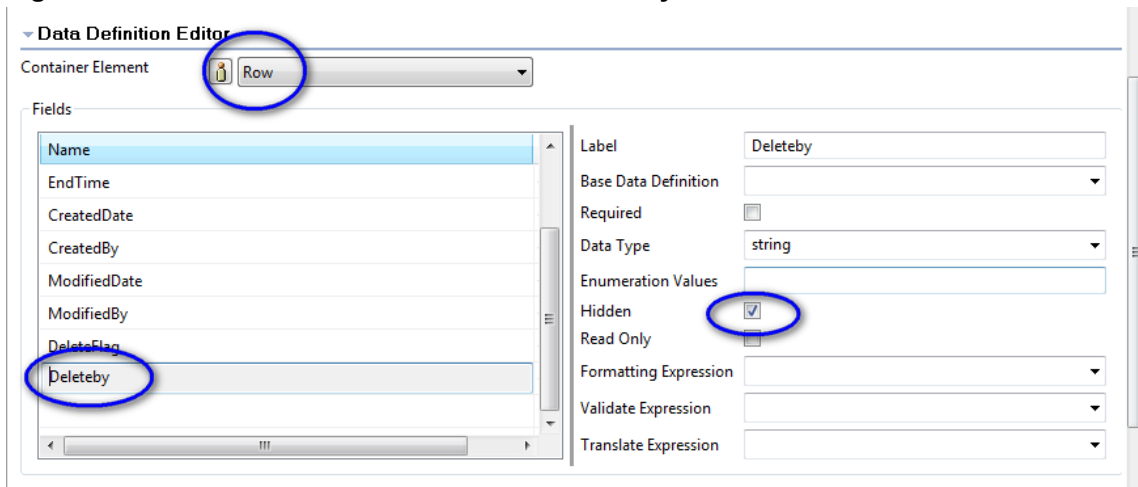
**Figure 96. Data Definition Editor section for DeleteFlag**



10. In the Data Definition Editor section, select Deleteby, updating the parameters with the following inputs value (see figure 97)

- Label: Deleteby
- Data Type: string
- Enabled "Hidden"

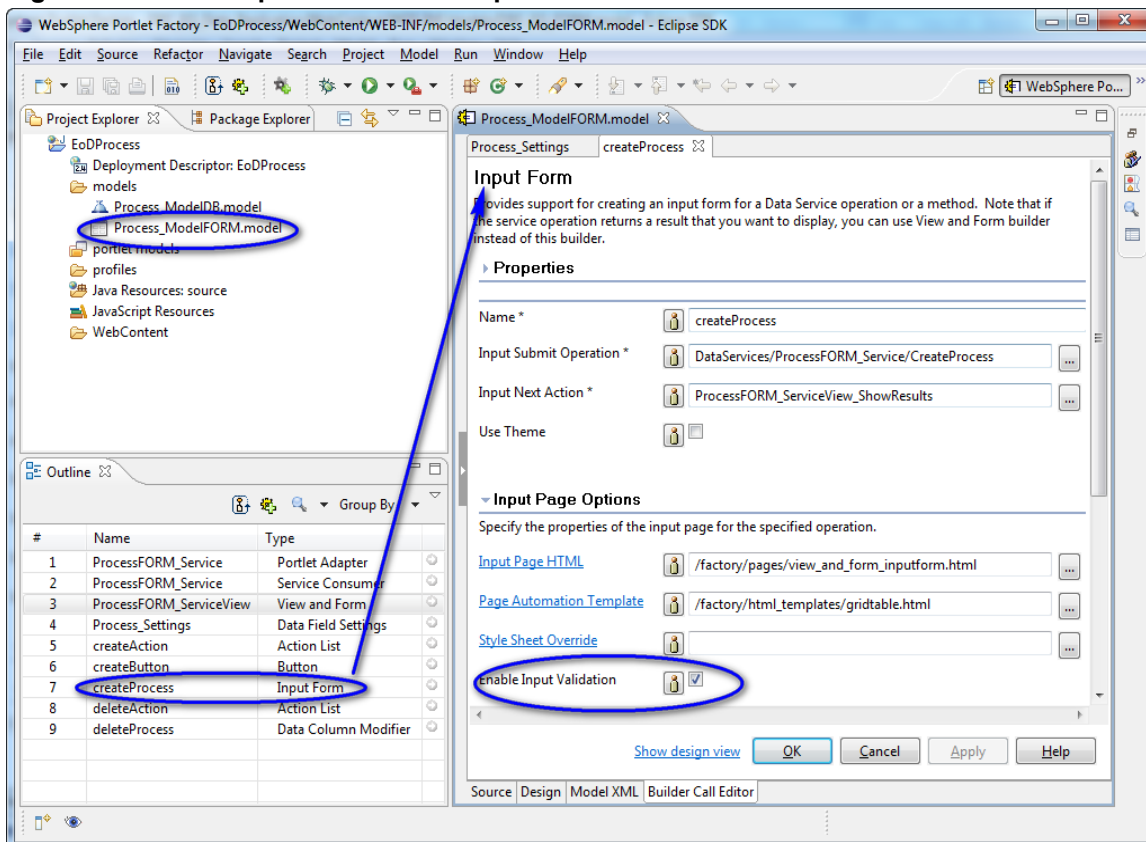
**Figure 97. Data Definition Editor section for Deleteby**



### 3.4 Enable input validation in input form

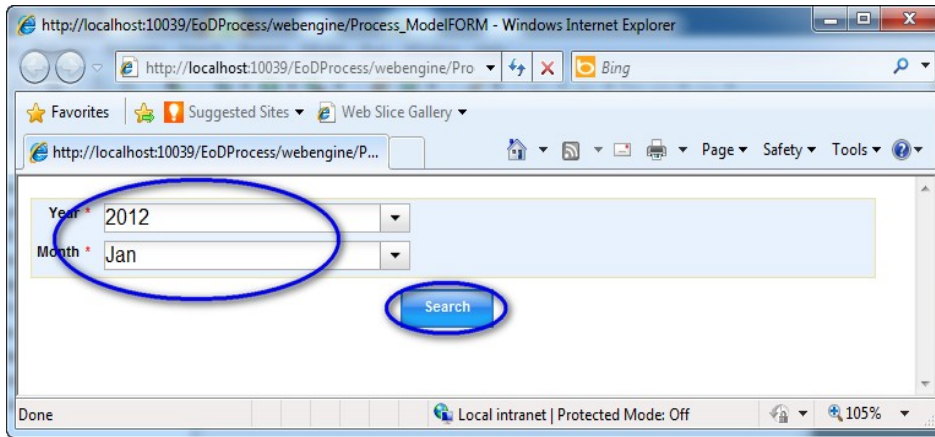
1. In the Project Explorer View, select Process\_ModelFORM, and then in the Outline View double-click the Builder Input Form createProcess (see figure 98). In the Input Page Options section, enable the Enable Input Validation option.

**Figure 98. Enable input validation in input form**



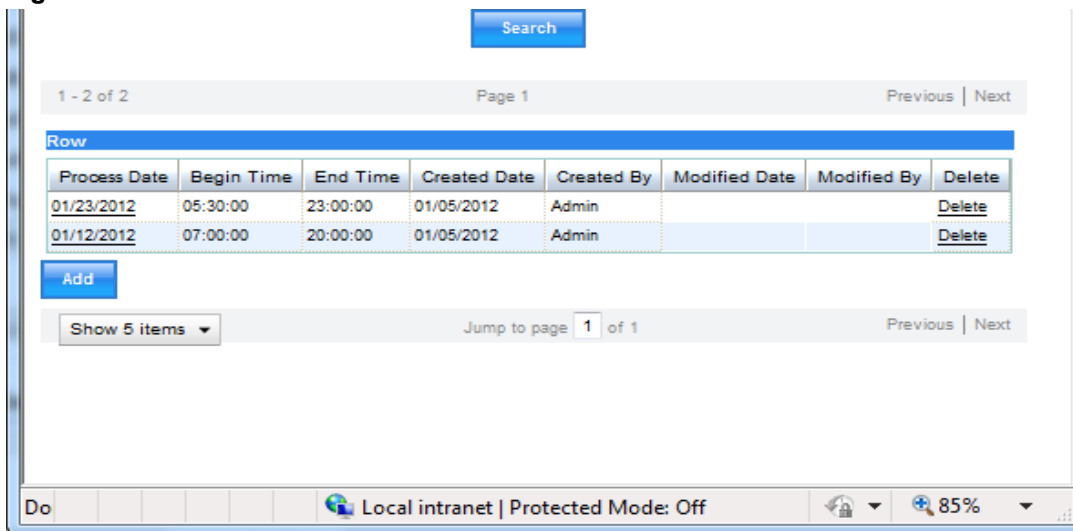
- To run the model, click the green arrow button, enter the Year and Month, and click the Search button, as shown in figure 99.

**Figure 99. Filter window**



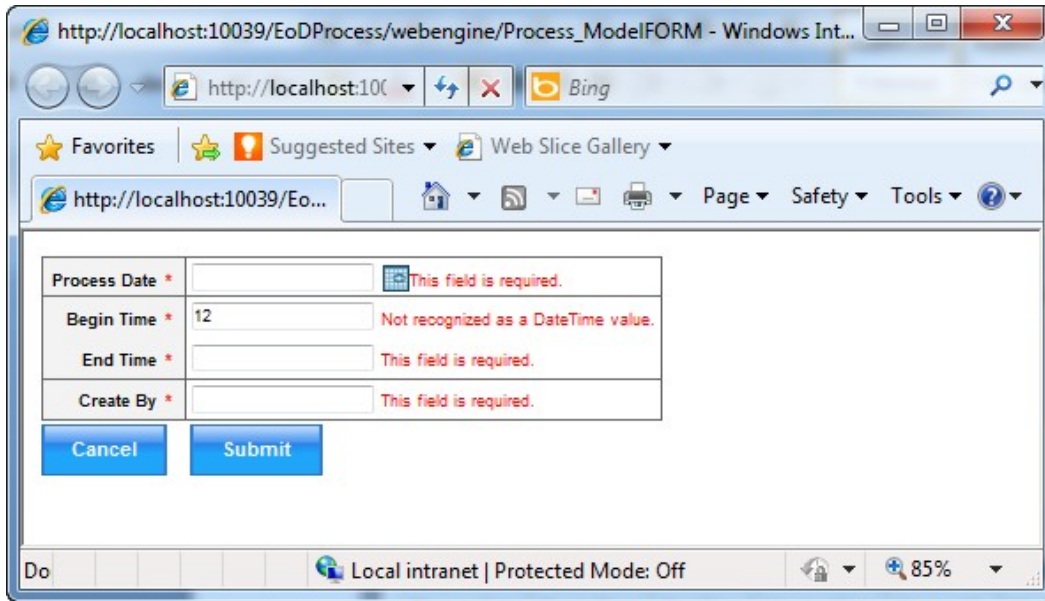
- Click the Add button, to add data (see figure 100).

**Figure 100. List window**



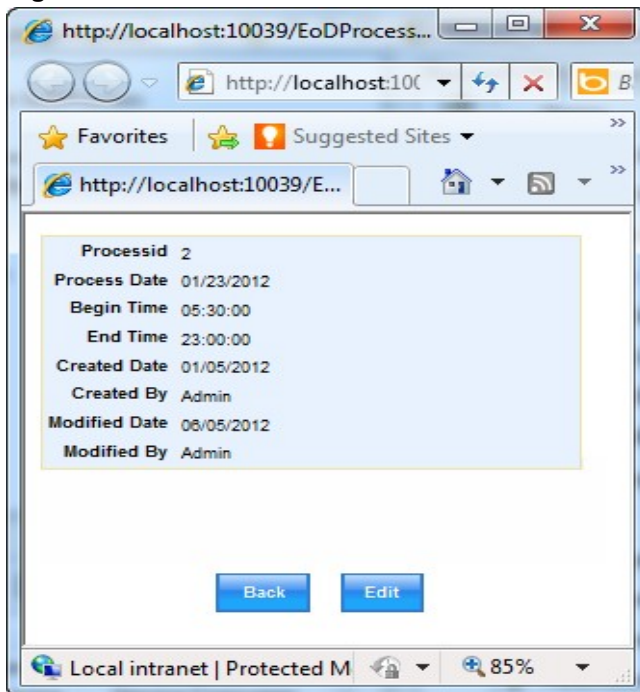
- Enter values in the Process Date, Begin Time, and End Time fields. Click the Submit button to save this data, or click the Cancel button to cancel, as shown in figure 101.

Figure 101. Add window



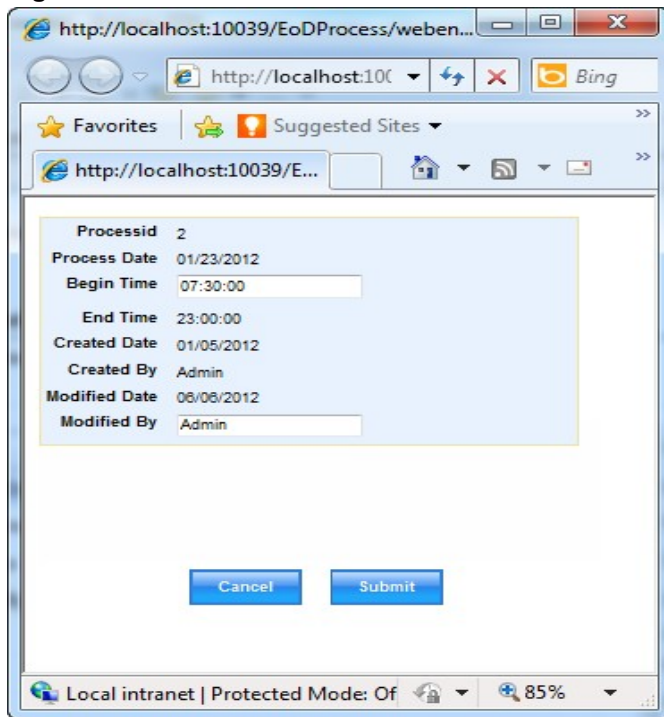
5. Click [Delete](#) to delete data, or click [01/23/2012](#) to view details (see figure 102).

Figure 102. Details window



6. Click the Edit button to edit this data, or click the Back button to go back to the list, and enter a value to edit. Click the Submit button to save this data, or click the Cancel button to cancel (see figure 103).

**Figure 103. Edit window**



## **4 Add notifications in the portlet input form**

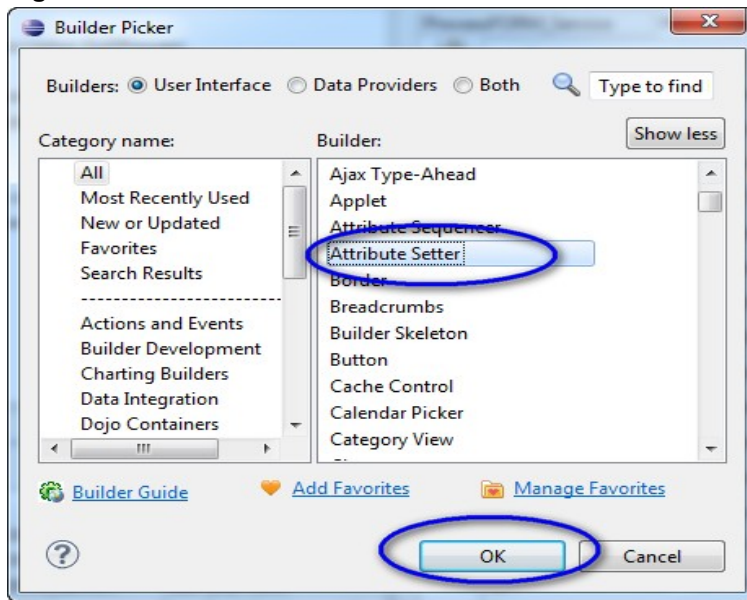
In this section we explain the steps for adding notifications in the portlet input form.

To begin the process, once again open the “EoDProcess” project we created in Section 1.

### **4.1 Add a delete notification**

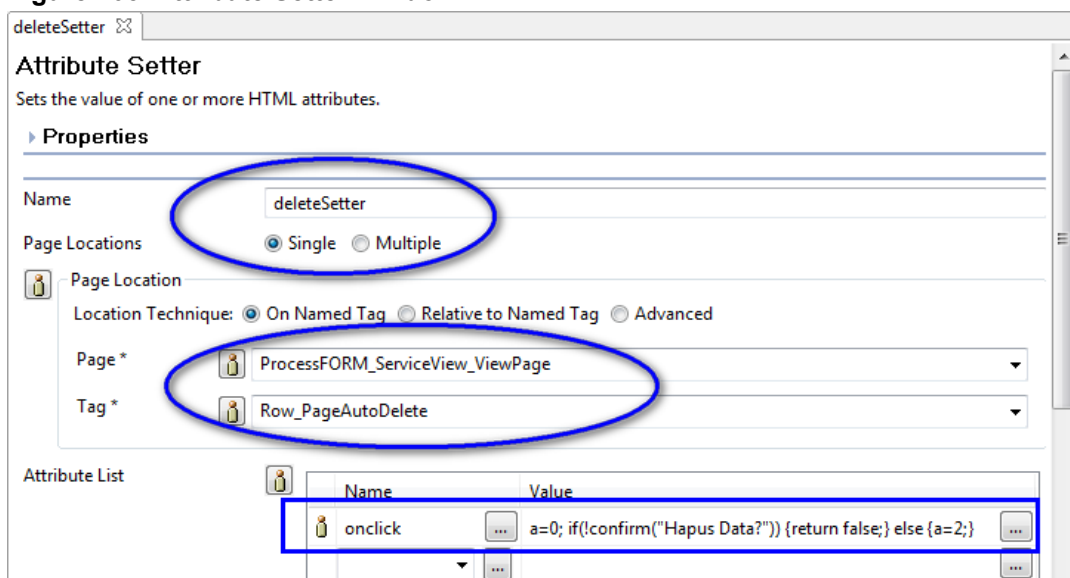
1. In the Outline View, click the builder picker icon and select “Attribute Setter” from the Builder list; click OK (see figure 104).

Figure 104. Attribute Setter selected



2. Enter deleteSetter in the Name field, select the “Single” option for the Page Location, and add the following inputs for these parameters (see figure 105):
  - Page: ProcessFORM\_ServiceView\_ViewPage
  - Tag: Row\_PageAutoDelete
  - Attribute List Name: onclick
  - Attribute List Value: a=0; if(!confirm("Are you sure to delete this data?")) {return false;} else {a=2;}
  - Overwrite Rule: Prepend new value
  - Value-less Attributes: Ignore the attribute

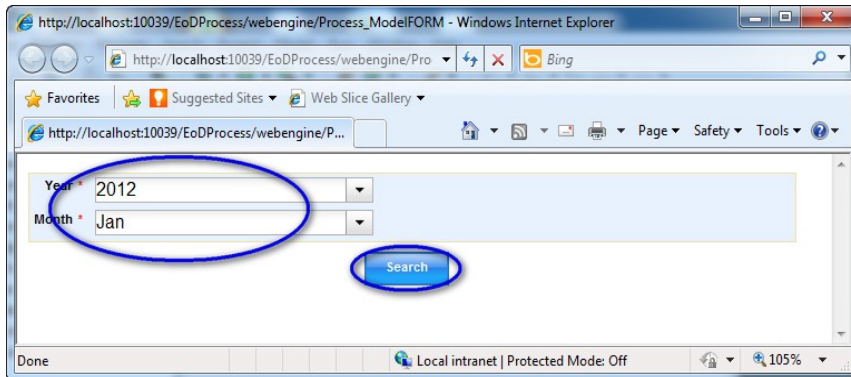
Figure 105. Attribute Setter Window





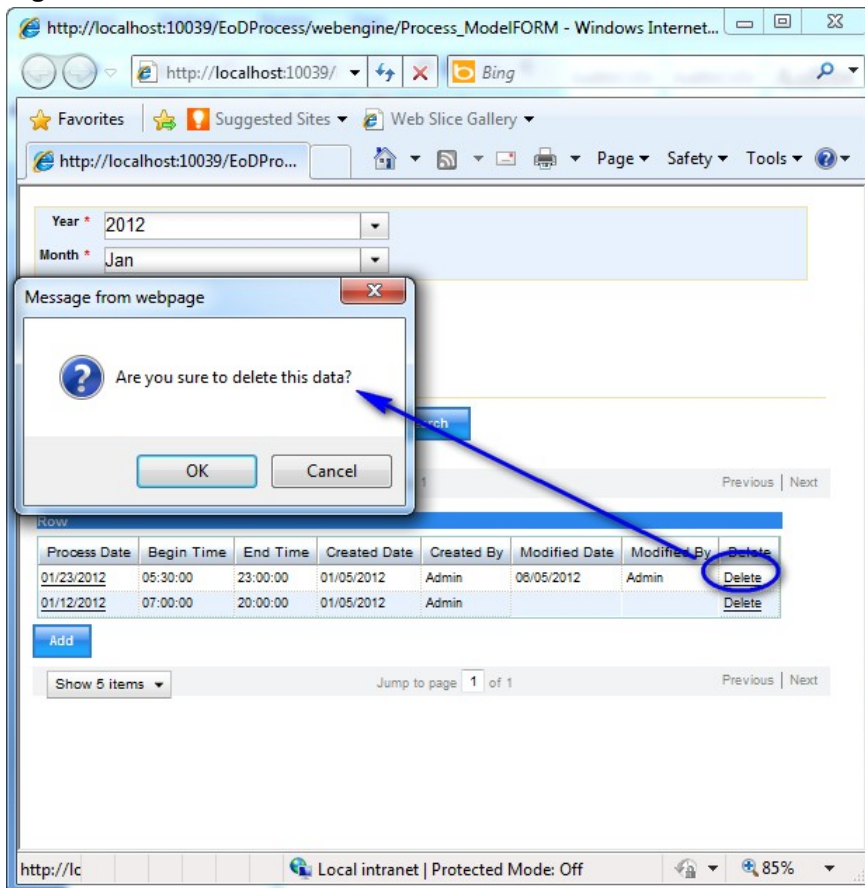
3. To run the model, click the green arrow button, enter the Year and Month, and then click Search (see figure 106).

**Figure 106. Filter window**



4. In the next window, click Delete; the notification will display as shown in figure 107.

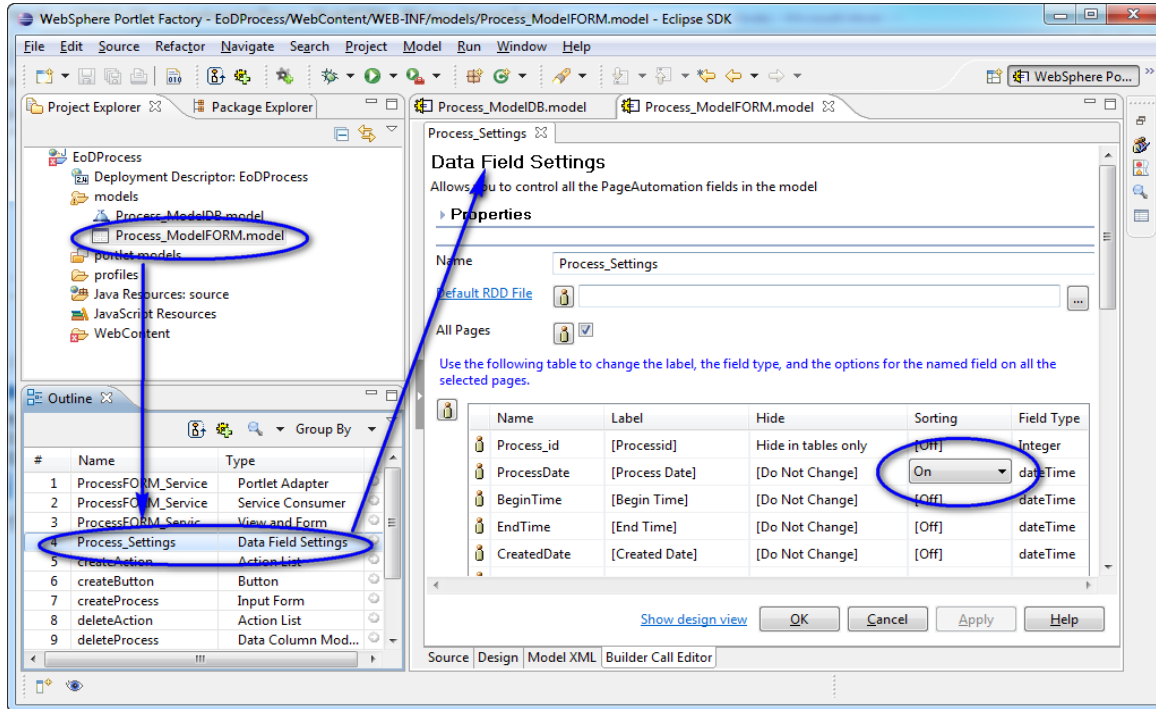
**Figure 107. Delete notification**



## 4.2 Sorting

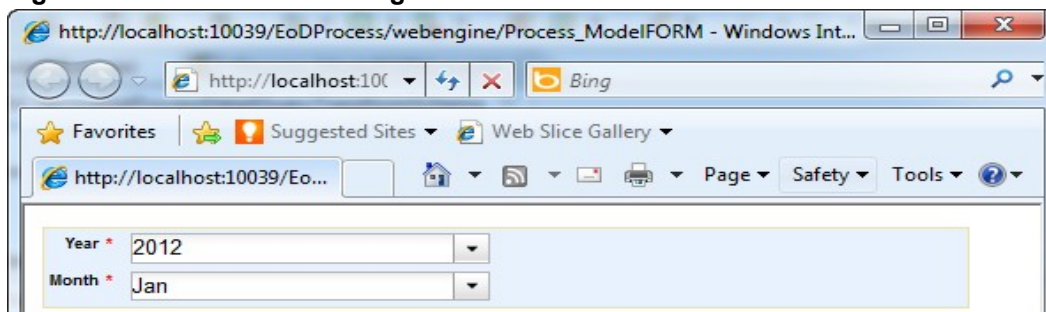
1. In the Project view, select Process\_ModelFORM, and then in the Outline View, double-click Data Field Setting Process\_Setting (see figure 108).

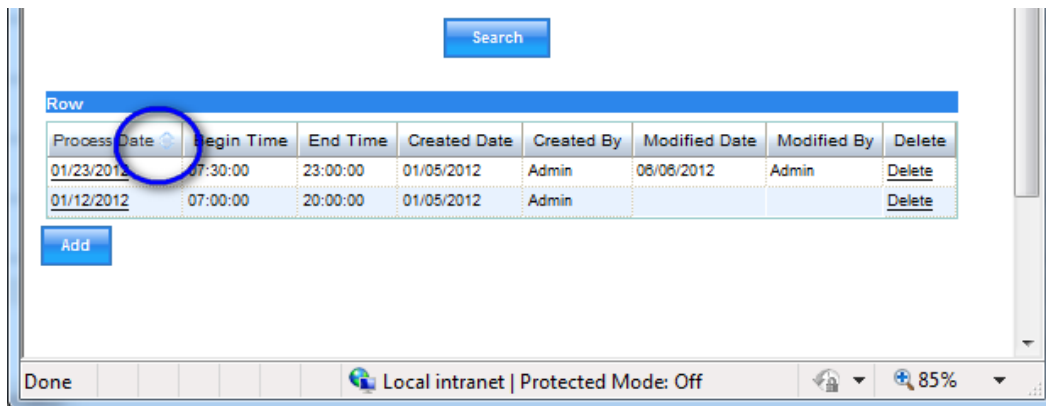
Figure 108. Sorting ProcessDate



2. To run the model, click the green arrow button, enter the Year and Month, and then click Process Date sorting on the List window(see figure 109).

Figure 109. List window sorting

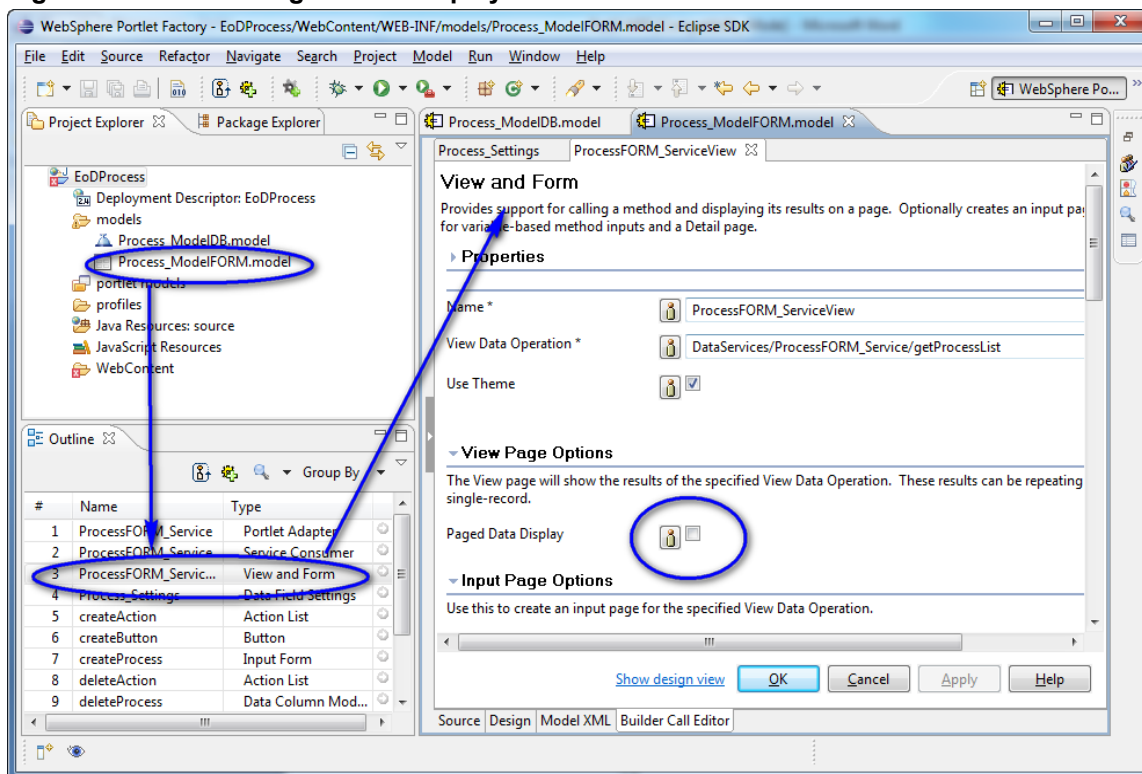




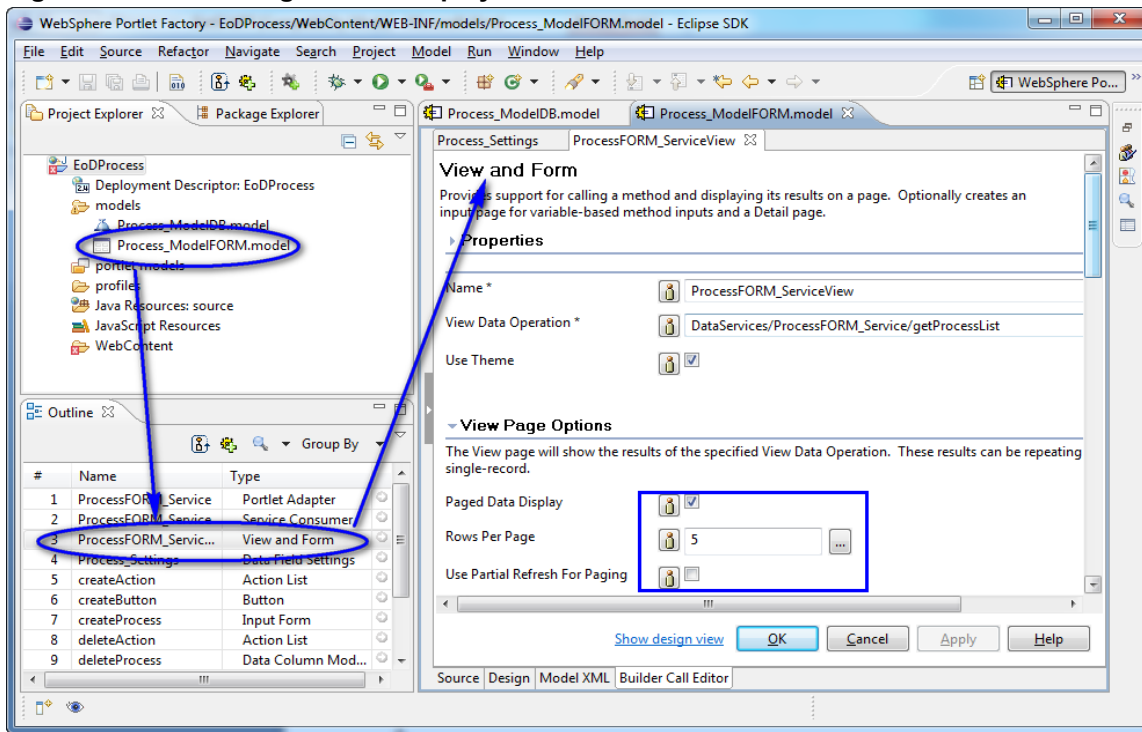
### 4.3 Enable/disable paging

1. In the Project view, select Process\_ModelFORM, and then in the Outline view, double-click ProcessFORM\_Service View and Form. See figures 110 and 111 for the disabled and enabled settings, respectively, in the View Page Options section.

Figure 110. Disable Paged Data Display



**Figure 111. Enable Paged Data Display**



2. To run the model, click the green arrow button, enter the Year and Month, and click Search. The List window with disabled paging and enabled paging displays in figures 112 and 113, respectively.

**Figure 112. List window with disabled paging**

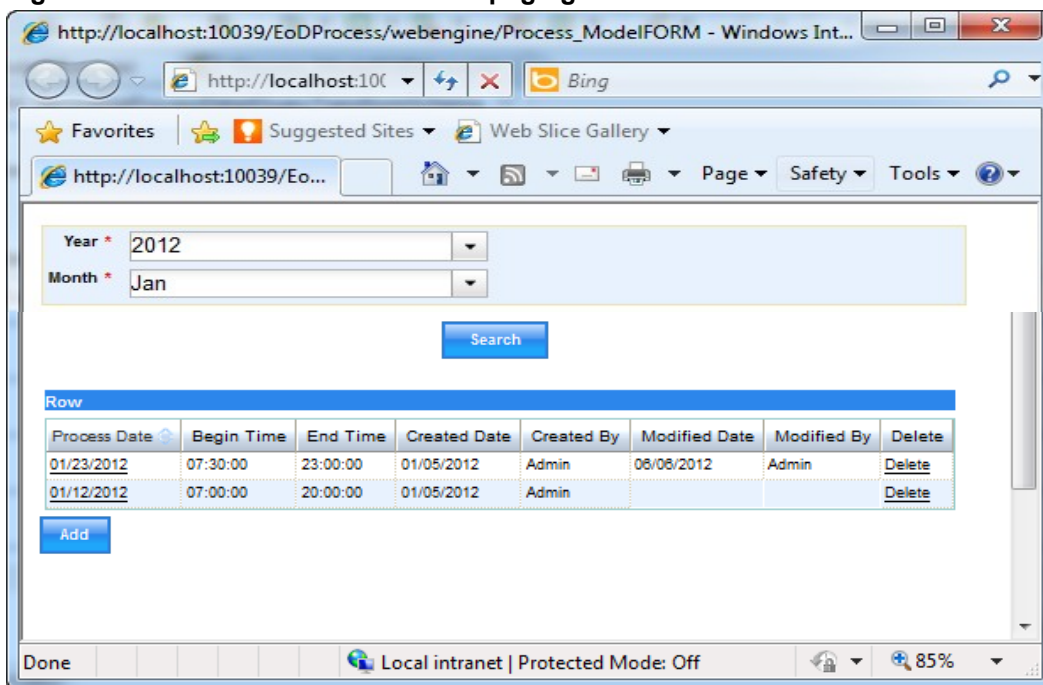
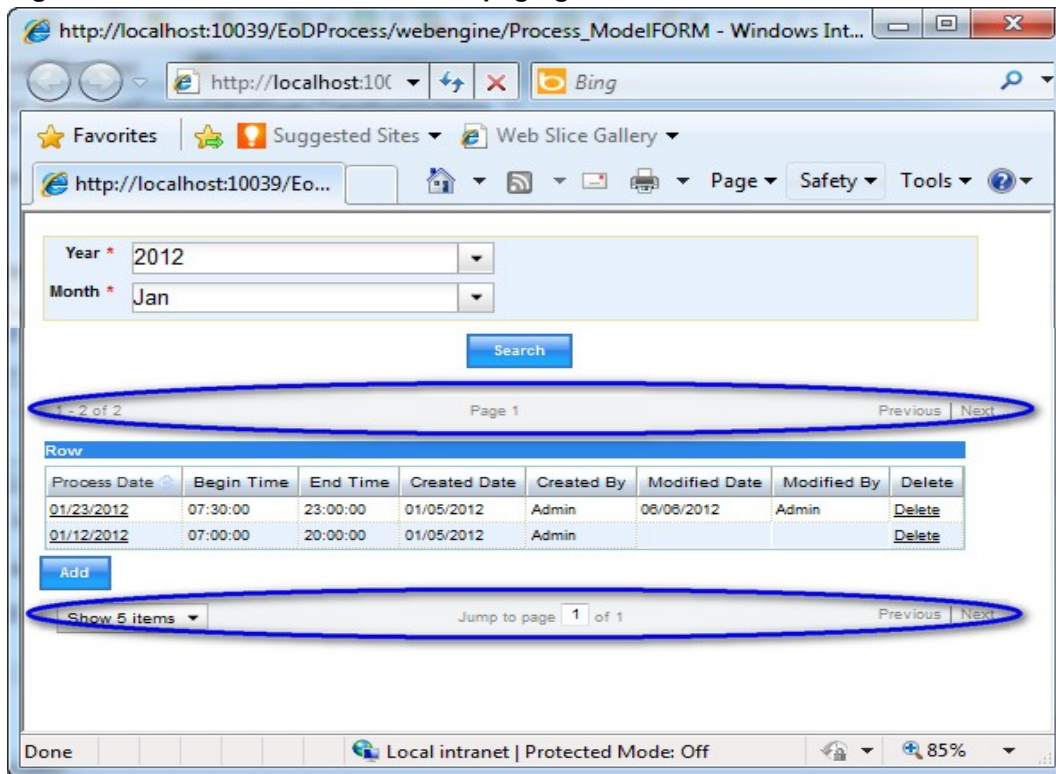


Figure 113. List window with enabled paging



## 5 Conclusion

This white paper has demonstrated how easy it is to create a simple, basic CRUD portlet with WebSphere Experience Factory and how to modify default fields generated from the wizard model with more advanced fields, like a combo box. We also explained the steps to add validation rules to the portlet and to add notifications in the portlet input form.

## 6 Download

The data definition language download file, DDL.txt, is for setting up the database. In our lab we used Microsoft® SQL Server, and you can use this script directly if you have an SQL Server database. If you use other databases, like Oracle or IBM DB2®, then minor modification is needed for the script to work.

## 7 Resources

- developerWorks® IBM Web Experience Factory product page:  
<http://www.ibm.com/developerworks/websphere/zones/portal/portletfactory/>
- IBM Web Experience Factory product documentation page:  
<http://www-10.lotus.com/ldd/pfwiki.nsf/xpViewCategories.xsp?lookupName=Product%20Documentation>
- WebSphere Application Server InfoCenter topic, “Configuring a data source using the administrative console”:

[http://pic.dhe.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=%2Fcom.ibm.websphere.base.doc%2Finfo%2Faes%2Fae%2Fdat\\_crrtpds.html](http://pic.dhe.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=%2Fcom.ibm.websphere.base.doc%2Finfo%2Faes%2Fae%2Fdat_crrtpds.html)

- IBM Web Experience Factory customer discussion forums:  
<http://www-304.ibm.com/support/docview.wss?rs=3044&uid=swg27011853>

## 8 Author biographies

**Priyoko T. Sudarmadi** leads the IBM WebSphere and WebSphere Portal team at PT. Anabatic Technologies in Jakarta, Indonesia. He works in Presales and as the Project Manager and Solution Architect for solutions based on the WebSphere brand and WebSphere Portal Family of products (Portal, IBM Lotus® Quickr®, IBM Sametime®, WEF, Dashboard Framework and IBM Connections).

**Vitri** is a Software Development Specialist on the WebSphere and Portal team at PT. Anabatic Technologies. She is certified as an IBM WebSphere Portal Solution Developer and is involved in numerous Portal projects that require extensive development and customization to meet client requirements.

**PT. Anabatic Technologies** is a leading Premier IBM Business Partner in Indonesia, with extensive implementation of IBM products in the areas of Social Collaboration, Portal, Data Warehouse, Business Intelligence, and Enterprise Content Management with many prominent companies in Indonesia.

### Trademarks

- DB2, developerWorks, IBM, Lotus, Quickr, Sametime, and WebSphere are trademarks or registered trademarks of IBM Corporation in the United States, other countries, or both.
- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Microsoft is a trademark of Microsoft Corporation in the United States, other countries, or both.
- Other company, product, or service names may be trademarks or service marks of others.