



Process Flow Design Pattern Scenarios

Scenario 1: Partner Data integration

Vendor XYZ manufactures travel gear and emails Product data to all of its retailers. The raw data file consists of Product fields such as product name, SKU, category quantity, price and description. This data is critical for the retailers and thus requires that their Website and back-end shipping and inventory applications are automatically updated with this product information.

As a developer the first step in developing this data flow is to gather additional technical requirements. The broad business objective (as explained above) provides just a guiding principle as to what the outcome is going to be. But it does not provide specifics as to 'How' or 'When' or 'What' to do in terms of achieving this objective.

As a best practice, one of the key elements is to extend this requirement and add integration specific rules.

Step 1: Technical Requirements

- Product Data arrives as an email attachment
- Product Data needs to be validated and exceptions filtered out
- Three target schemas are going to be used in Mapping to generate three output data streams.
- Valid data is sent to a database application, email attachment and to a Web Site

Step 2: Design

This document assumes that the relevant Activities have already been created. We will focus on how to design the solution in Adeptia BPM's Process Designer.

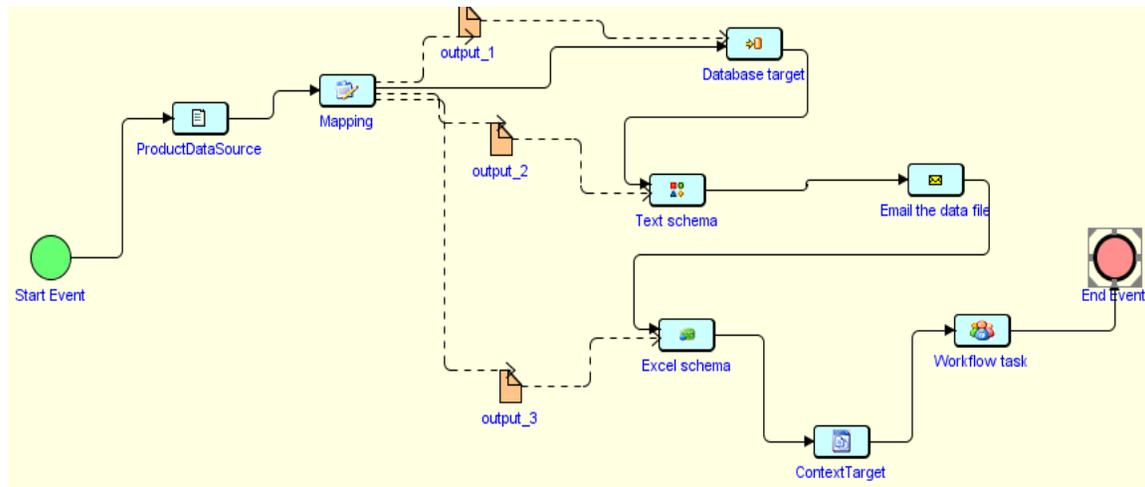
In the canvas area we will bring all the pre-defined activities:

- First activity after the Start Event is the Email Source. Logically the first step during the execution of the process flow is to get the data file from the Vendor.
- Second activity is the Mapping activity. Please note that if the Source data file is XML then no Schema activity is needed in the second step of this flow.

If the Source data is XML or Database then we will just place the Mapping activity as the second activity in the flow.

- Third activity is the Database Target.
- Fourth activity is the related schemas objects for the desired output data format that we are generating. Make sure that the Transformer type (in their property panel) is set to 'XMLtoStream' for all of these Target schema activities.
- Now we will connect these activities together. Please note that we will use Multiple Stream option (by right-clicking on the Mapping activity) to map the three data streams to the appropriate Activity. Thus the output_1 of the Mapping is mapped to Database target, and the rest are mapped to the particular Target schemas.
- One of the Targets is a Workflow Task which will allow User to download the Error records (output 3 of the mapping). We will use Context Target node before the Workflow task. This Context Target stores the excel output file locally which allows us to provide the link in the Workflow task form.

The end result of the design should be as follows:



Scenario 2: Purchase Order automation

Now let's look at how various sub-processes can be used to build a business process.

Step 1: Technical Requirements

We will take the Purchase Order Automation as an example, which consists of:

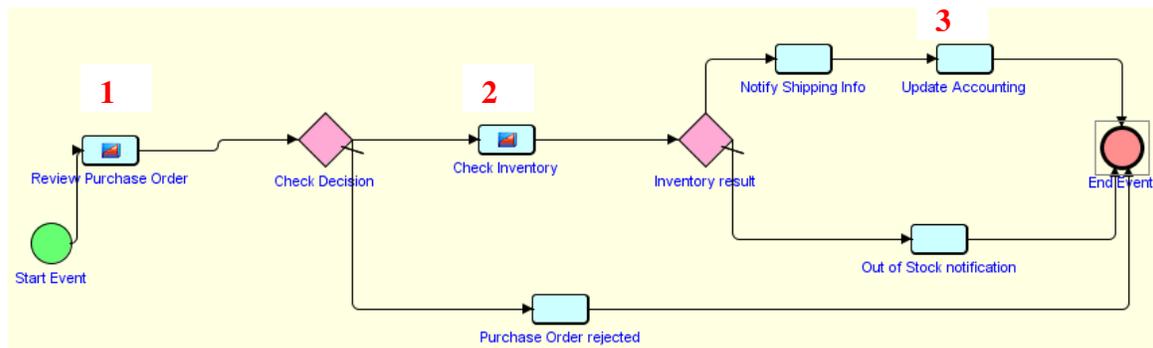
- Sub process # 1: Review Purchase Order
 - o Receive PO from an email

- Show it in the Workflow Task (assigned to a User)
 - Make a decision on the PO, and pass the Decision value to the next sub-process
- Sub process # 2: Check Inventory
 - If the PO is approved then check inventory in the database based on the Product ID
 - If the value is True then automatically notify the Customer about the shipping and delivery information
 - If False then notify the Customer that the product is out-of-stock
- Sub process # 3: Update Accounting
 - If the Inventory Check returns True then update the Accounting system. This will take the Unit Cost and Order Quantity and insert this information in a database
 - Notify Accounting Manager about the PO

Step 2: Design

As described above, this Business Process has three sub-processes and each of these sub-processes consist of data flows that are called by the parent Business Process.

Following is the Business Process Flow Design. We can also term this flow as the 'Parent' process flow. There are three sub-processes marked in red.



Nodes marked 1, 2, and 3 in the design are Call actions that execute the associated sub-processes.

- For example, Node 1 is labeled as 'Review Purchase Order' which executes the data flow that performs the steps described for Sub Process # 1 above.
- Call actions require User to select which particular process flow to execute. These are synchronous calls and the next activities in the parent process flow (such as the one depicted above) are only executed once this sub-process execution is completed.

Image below represents the properties of the first Call action 'Review Purchase Order'. The flowid field displays the specific sub-process flow 'Review PO' that this Call action is executing.

Name	Value
flowId	Review PO
Label	Review Purchase Order
Name	Call1
resultCtxVarName	
Type	Call

To bring a value from this child process 'Review PO' into the context of the parent process flow, we have to use the Set-Parent-Context Action in this sub-process.

Name	Value
Activity	
childActivityName	
ChildKey	
ChildName	
Key	
Label	SetChildContext5
Name	SetChildContext5
Type	Set-Child-Context

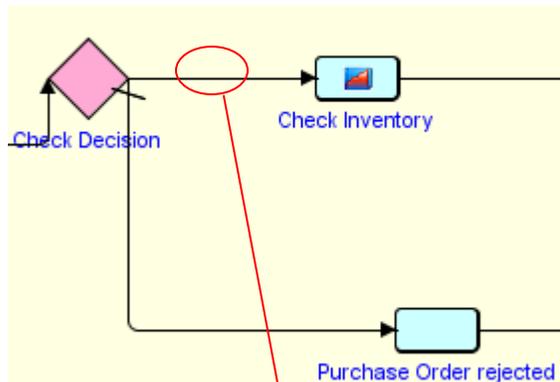
In the box above:

- **Activity:** Name of activity of the child process flow, whose 'Activity Context Variable' value specified by Key will be used to set the parent process flow variable specified by Parentkey. If name of the activity is not specified, then the 'Process Flow Context Variable' specified by Key will be used.
- **Key:** Name of the context variable of the child process flow whose value will be used to set the parent context variable specified by ParentKey.
- **Label:** The label displayed for the Set-Parent-Context action.
- **Name:** Name of Set-Parent-Context action
- **ParentActivityName:** Name of activity of parent process flow whose 'Activity Context Variable' specified by ParentKey will be set. If name of the activity is not specified, then the 'Process Flow Context Variable' specified by ParentKey will be set.
- **ParentKey:** Name of Context Variable of parent process flow whose value will be set.

For example, we have a variable in the child process named as 'reviewDecision'. Now another variable with the same name is also set in the parent process flow. When this child process is executed then its variable value is set and then passed to the parent's variable 'reviewDecision'. Thus in the box above we will enter the variable name in the ChildKey and in the ParentKey field. At run-time, this Set-Parent-Context will assign the value to the parent flow's variable.

If we need to assign a value to the child process flow's variable so that we can use it during the execution of the sub-process, then we will use the Set-Child-Context function which has similar properties.

Now once we have brought the child process's variable value into the context of the parent process flow, we can apply a rule on this value in our Gateway node. Double-click on the control arrow going from the Gateway 'Check Decision' to 'Check Inventory' activity, and then in the Condition Wizard window configure the following rule.



Rule Information

Rule Name: Rule Reason:

Declaring Activity Attribute/variable to build Expression
Please select Activity Attribute/Variable and desired value to build expression:

Existing Activities :	<input type="text"/>
Activity Attribute/Variable :	<input type="text" value="reviewDecision"/>
Operand Type :	<input type="text" value="Text"/>
Operator :	<input type="text" value="Equal"/>
Value :	<input type="text" value="Approved"/>

Declaring Operator to build Expression
Please select required Operator to build Expression:

Select Operator:

Expression

```
{ $$reviewDecision$$ .equals ("Approved" ) }
```

If the reviewDecision equals Approved, then the flow order from the Gateway is routed to the Check Inventory path. Else an email notification informing the customer about the rejection of the Purchase Order is sent by the 'Purchase Order rejected' activity.