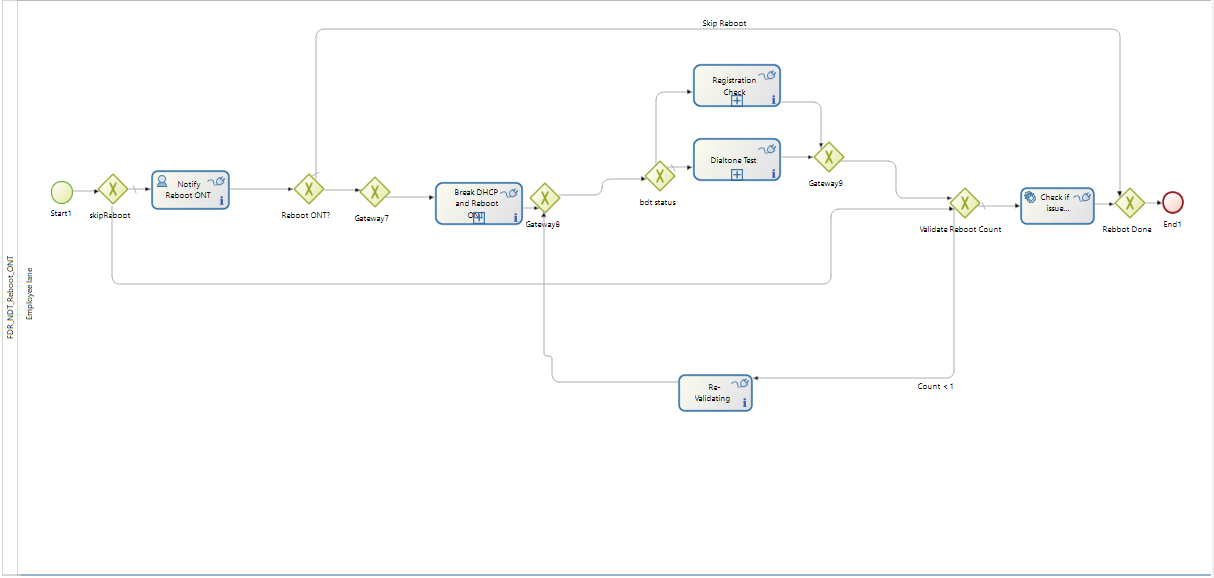
**Bonita to PAM Migration – Technical Implementation of** FDR\_NDT\_Reboot\_ONT **Workflow: Version:** 1.0.1.9.2



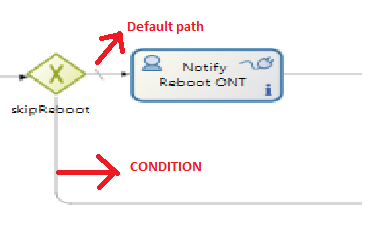
Process Variables:

|  |  |
| --- | --- |
| actionItems | (Java\_object) |
| actionId | (Text) |
| breakDialToneStatus | (Text) |
| ckttype | (Text) |
| customertopology | (Text) |
| command | (Text) |
| dataPortStatus | (Boolean) |
| description | (Text) |
| dialToneIssueResolved | (Boolean) |
| eltype | (Text) |
| finalActionResult | (Text) |
| hasGoodDialTone | (Boolean) |
| hasGoodPing | (Boolean) |
| hasSipPortIssues | (Boolean) |
| inLineId | (Text) |
| initialBreakDialToneStatus | (Text) |
| isDataPortUp | (Boolean) |
| isManualRebootDone | (Boolean) |
| isParallel | (Boolean) |
| isPortUp | (Boolean) |
| isStatic | (Boolean) |
| issueResolved | (Boolean) |
| l3ICMPPingStatus | (Text) |
| manualFlag | (Integer) |
| manualTaskResult | (Text) |
| microserviceurl | (Text) |
| newRegDate | (Java\_object) |
| ontType | (Text) |
| ontTypePub | (Text) |
| ontVendorType | (Text) |
| pingBHRMap | (Text) |
| pingStatus | (Text) |
| portType | (Text) |
| registrationDate | (Java\_object) |
| registrationStatus | (Text) |
| responseString | (Text) |
| resultsetpublic | (Java\_object) |
| sipAlarmStatus | (Text) |
| sipAlarmTroubleCode | (Text) |
| skipVoipPortCheck | (Boolean) |
| title | (Text) |
| transactionid | (Text) |
| troubletype | (Text) |
| validateCounter | (Integer) |
| voiceServiceStatus | (Boolean) |

**Steps to be implemented**:

->Start node will call skipReboot (XOR Gateway). It will have two decision paths in which one would be the default gateway and other will have a condition on the connector from skipReboot to Validate Reboot Count.

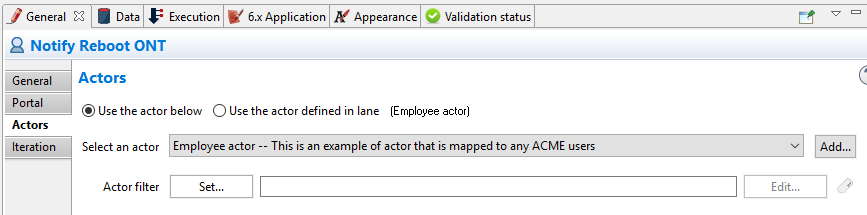




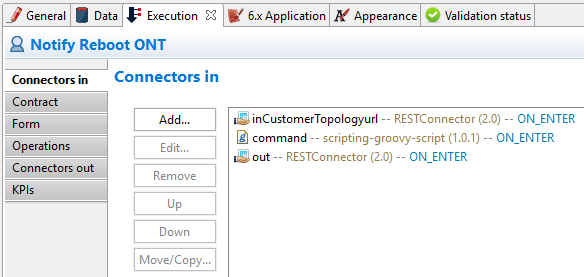
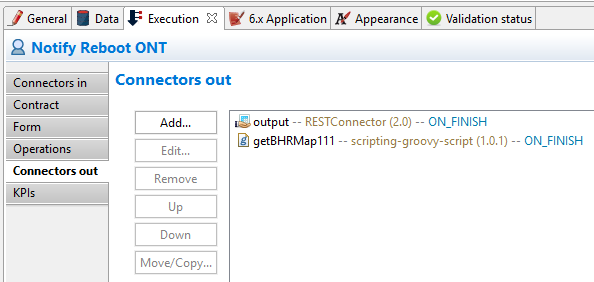
|  |  |
| --- | --- |
| Condition | Default |

|  |  |
| --- | --- |
| Condition | isManualRebootDone || "MDU".equalsIgnoreCase(ontType) |

Default condition path will reach to Human task node (Notify Reboot ONT). It will have an Actor assigned to it when you select general properties of **Notify Reboot ONT** node as shown below:



Also in the Execution properties pane we can see Connectors in and Connectors out on the node as shown below:

**Connectors in**

inCustomerTopologyurl (Id: inCustomerTopologyurl, Type: RESTConnector, Event: ON\_ENTER, Ignore errors: false)

command (Id: command, Type: scripting-groovy-script, Event: ON\_ENTER, Ignore errors: false)

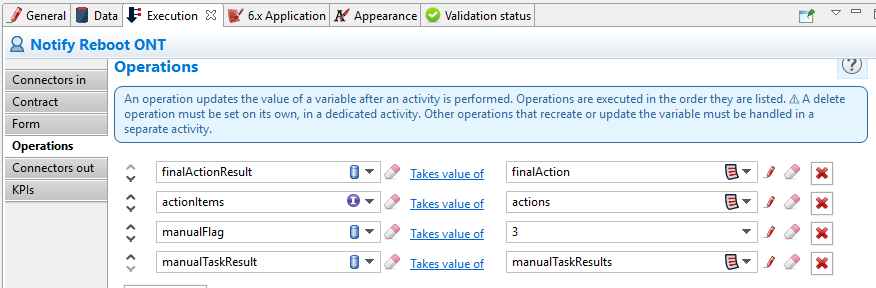
out (Id: out, Type: RESTConnector, Event: ON\_ENTER, Ignore errors: false)

**Connectors out**

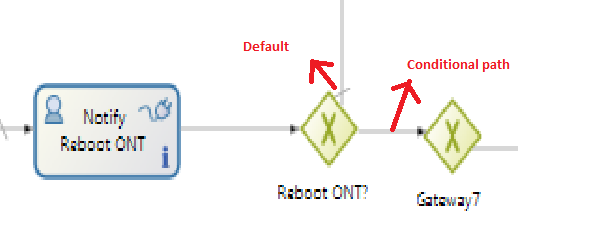
Output (Id: output, Type: RESTConnector, Event: ON\_FINISH, Ignore errors: false)

getBHRMap111 (Id: getBHRMap111, Type: scripting-groovy-script, Event: ON\_FINISH, Ignore errors: false)

Also there are operations that needs to be done on this node and assign values to the variables accordingly as shown below:

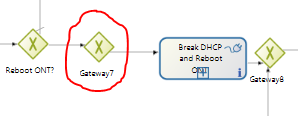


After Notify Reboot ONT, it reaches to **Reboot ONT?** Gateway node.



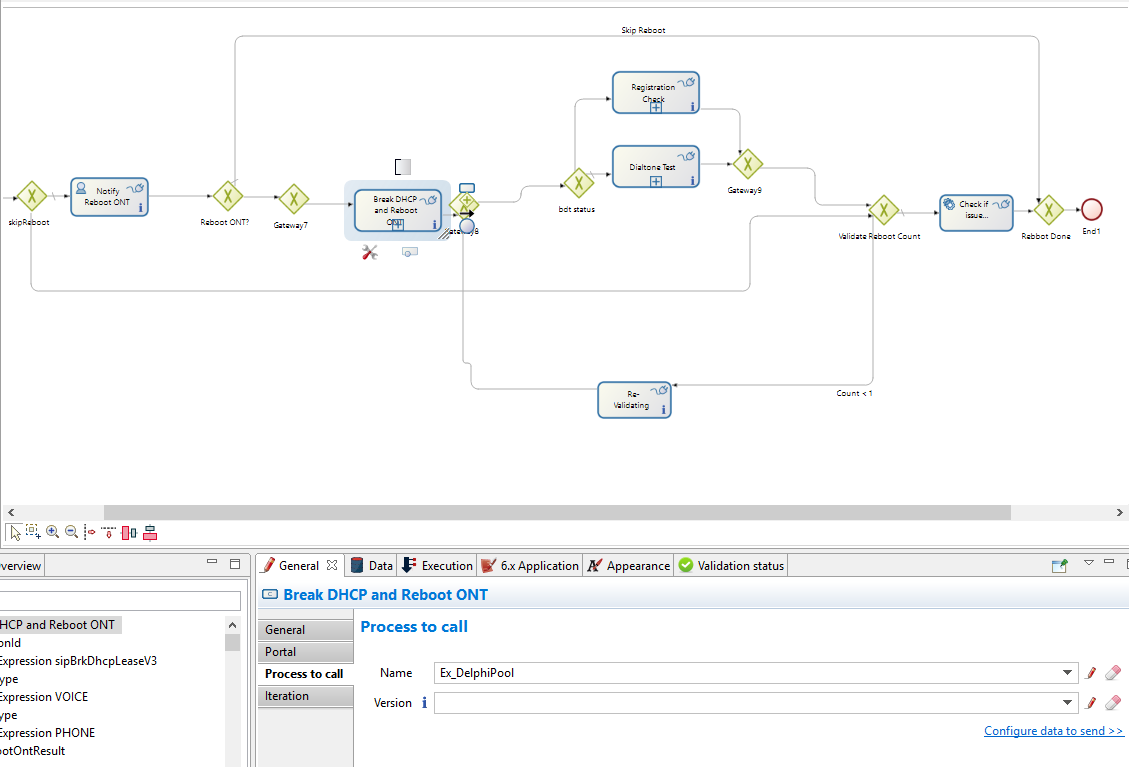
|  |  |
| --- | --- |
| Condition | Default |

|  |  |
| --- | --- |
| Condition | return finalActionResult.toUpperCase().contains("REBOOT ONT") |



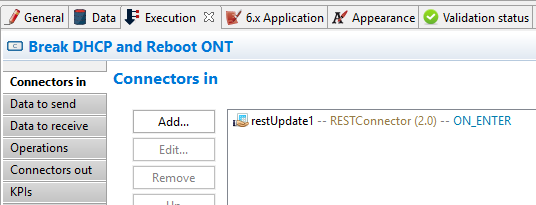
\*\*As shown above, we don’t need Gateway7 (Immediate XOR Gateway without any conditional paths wouldn’t serve any purpose). So we will eliminate the Gateway7 from flow and connect directly from **RebootONT?** To **Break DHCP and Reboot ONT.**

**Break DHCP and Reboot ONT** is a reusable sub-process and it calls **Ex\_DelphiPool** workflow. Also it has Connectors in, Connectors out and Operations on the node.



**Connectors in**

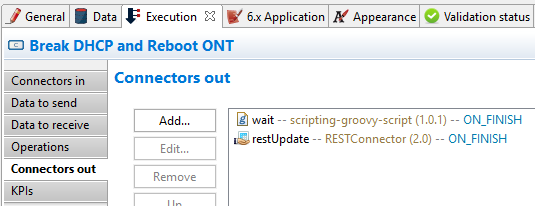
restUpdate1 (Id: restUpdate1, Type: RESTConnector, Event: ON\_ENTER, Ignore errors: false)



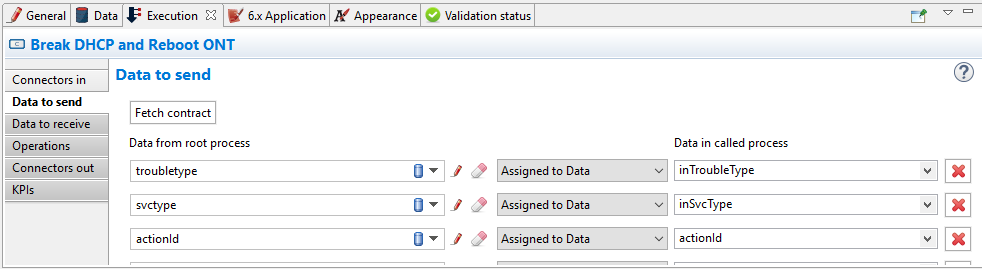
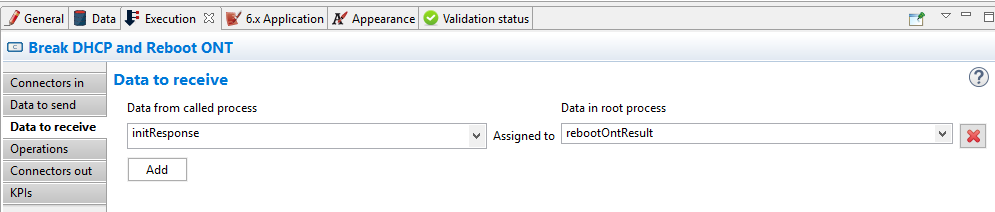
**Connectors out**

wait (Id: wait, Type: scripting-groovy-script, Event: ON\_FINISH, Ignore errors: false)

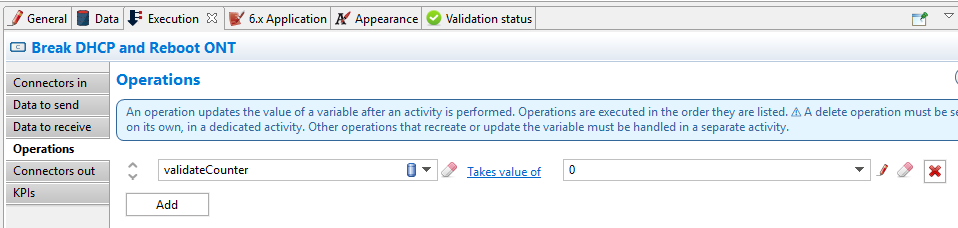
restUpdate (Id: restUpdate, Type: RESTConnector, Event: ON\_FINISH, Ignore errors: false)



**Data to send and receive** from **Sub-process** has to be mapped as shown below:

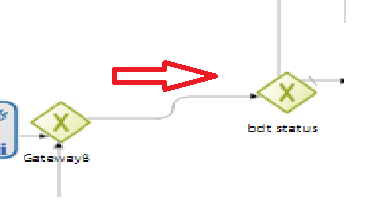
 

Operations on the node are to be assigned after the completion of the activity as shown below:

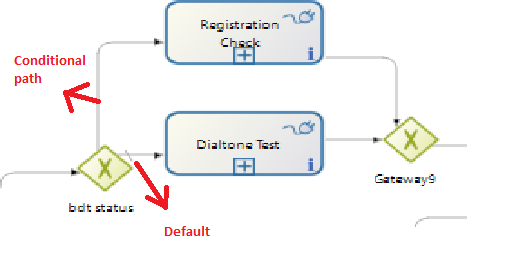


After that **Break DHCP and Reboot ONT.** It reaches **Gateway 8.**

**Gateway 8 acts here as converging gateway. Finally, it will be reaching to bdt status XOR gateway.**



From the **bdt status** there are two decision paths. Default one will be to **Dialtone Test** Subprocess and conditioned path would be to **Registration check**.

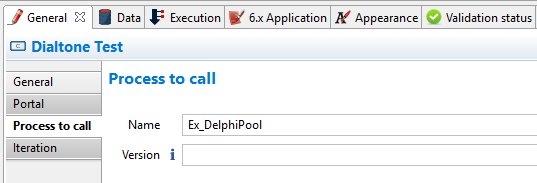


|  |  |
| --- | --- |
| Condition | Default |

|  |  |
| --- | --- |
| Condition | return "NOTSUPPORTED".equalsIgnoreCase(initialBreakDialToneStatus) |

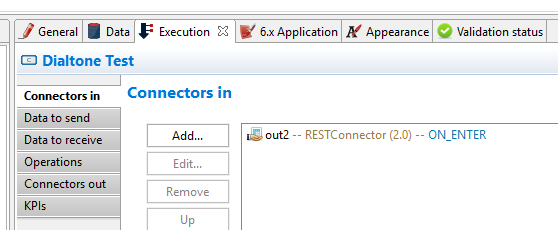
* **Dialtone Test (Reusable Subprocess):**

**“Dialtone Test”** is a reusable sub-process and it calls **Ex\_DelphiPool** workflow. Also it has Connectors in, Connectors out and Operations on the node.



**Connectors in**

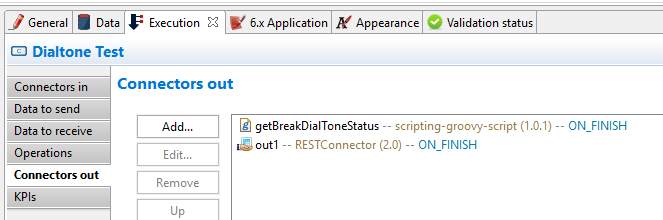
**Out2** (Id: out2, Type: RESTConnector, Event: ON\_ENTER, Ignore errors: false)



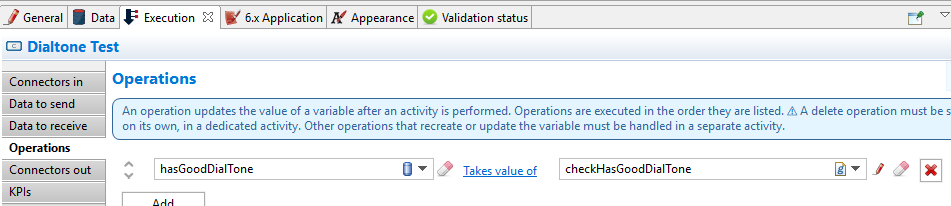
**Connectors out**

**getBreakDialToneStatus** (Id: getBreakDialToneStatus, Type: scripting-groovy-script, Event: ON\_FINISH, Ignore errors: false)

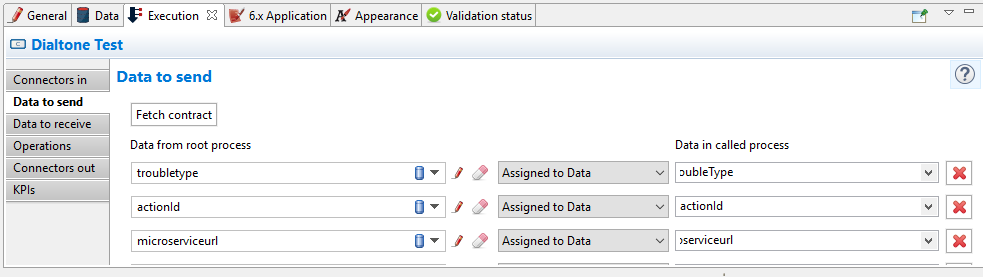
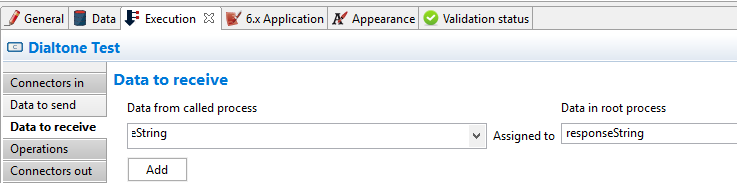
**out1** (Id: out1, Type: RESTConnector, Event: ON\_FINISH, Ignore errors: false)



Operations on the node are to be assigned after the completion of the activity as shown below:

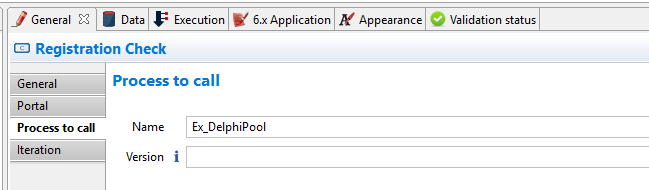


**Data to send and receive** from **Sub-process** has to be mapped as shown below:

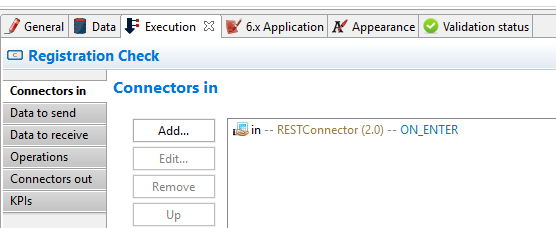
**Registration Check** (**Reusable Subprocess**):

**“Registration Check”** is a reusable sub-process and it calls **Ex\_DelphiPool** workflow. Also it has Connectors in, Connectors out and Operations on the node.



**Connectors in**

**in (**Id: in, Type: RESTConnector, Event: ON\_ENTER, Ignore errors: false**)**

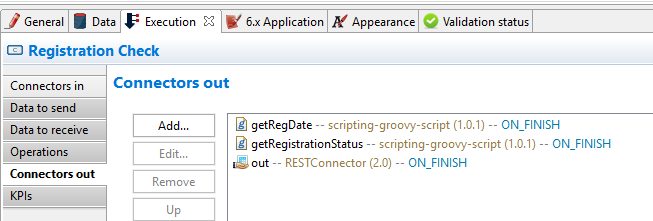


**Connectors out**

**getRegDate (**Id: getRegDate, Type: scripting-groovy-script, Event: ON\_FINISH, Ignore errors: false**)**

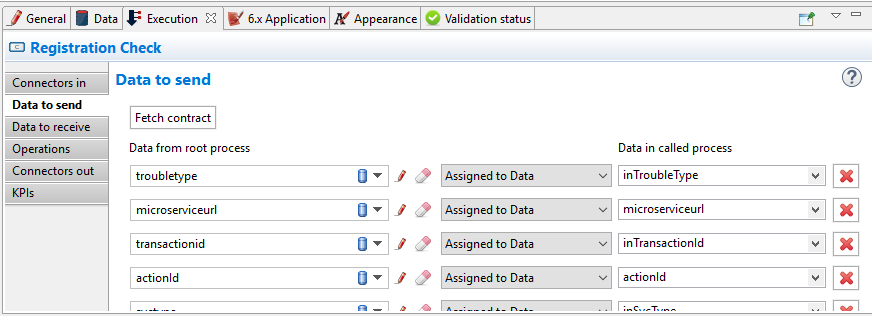
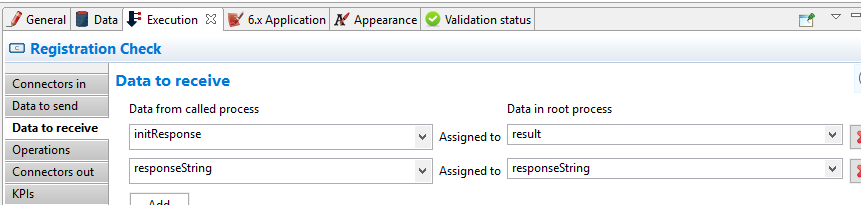
**getRegistrationStatus** (Id: getRegistrationStatus Type: scripting-groovy-script Event: ON\_FINISH Ignore errors: false)

**out (**Id: out, Type: RESTConnector, Event: ON\_ENTER, Ignore errors: false**)**

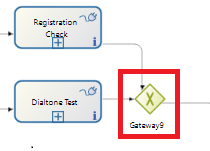


**Operations: There are no operations performed on this node.**

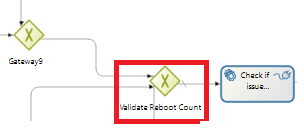
**Data to send and receive** from **Sub-process** has to be mapped as shown below:

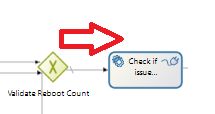
After the execution of Registration Check or Dialtone Test it will reach Gateway9(XOR Gateway).



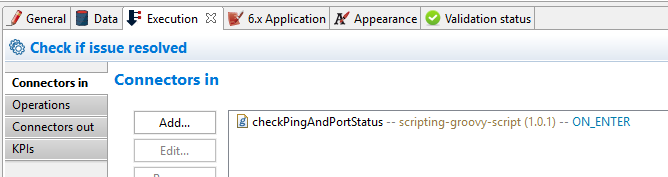
From gateway9 it moves to **Validate Reboot Count** flow node which is XOR gateway, In this case it acts as Converging.



First, we will walk through default path from **Validate Reboot Count:**



Then it will reach “**Check if issue resolved”** Service task and it has one connector on it.

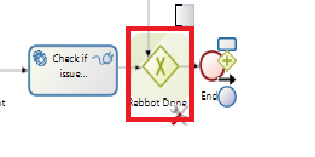


**Connectors in**

**checkPingAndPortStatus (**Id: checkPingAndPortStatus, Type: scripting-groovy-script, Event: ON\_ENTER, Ignore errors: false**)**

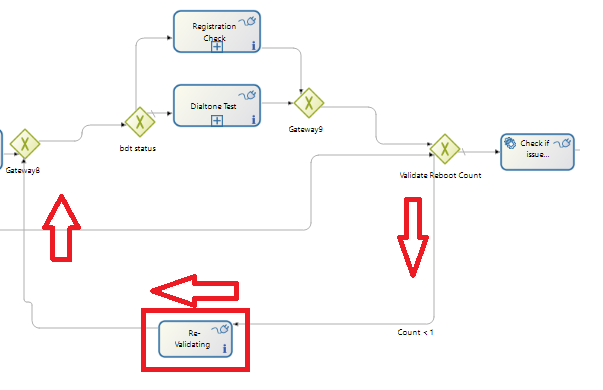
**Operations: There are no operations performed on this node.**

Then it will move to next node in the flow i.e. **Reboot Done (XOR gateway).** In this case this gateway acts as converging gateway, no need to specify any conditions on the connectors.



And Flow will be completed as shown above.

On the other hand there is another conditional path from the **Validate Reboot Count** as shown below.

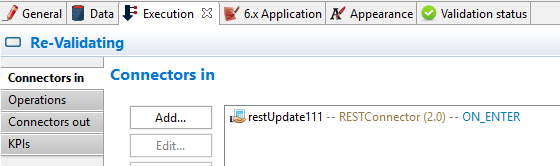
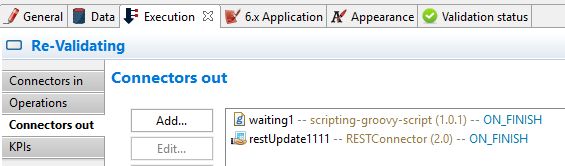


The condition specified on the connector from **Validate Reboot Count** as below.

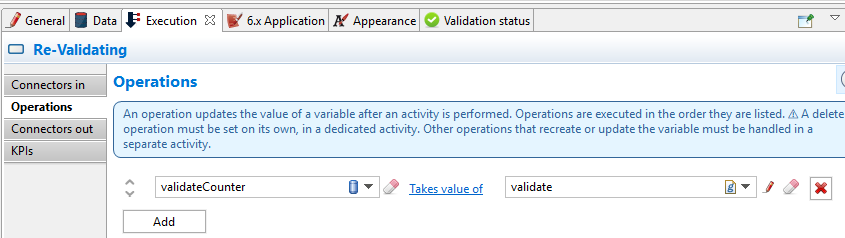
|  |  |
| --- | --- |
| Condition | //return validateCounter < 1 && !hasGoodPing && finalActionResult.toUpperCase().contains("REBOOT ONT") try{ return validateCounter < 1 && !dialToneIssueResolved && finalActionResult.toUpperCase().contains("REBOOT ONT") && ! ("SUCCESS".equalsIgnoreCase(registrationStatus)) } catch(Exception e){ return false; }/ \*if(skipVoipPortCheck){ return validateCounter < 1 && !dialToneIssueResolved && finalActionResult.toUpperCase().contains("REBOOT ONT") }else{ return validateCounter < 1 && ! dialToneIssueResolved && !isPortUp && finalActionResult.toUpperCase().contains("REBOOT ONT") } \*/ |

It will then reach to the flow node: **ReValidating**:

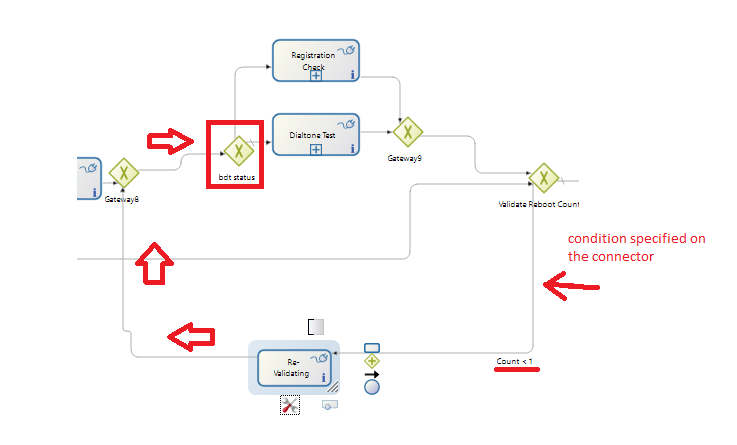
**Connectors in** and **connectors out** for this node are as follows:

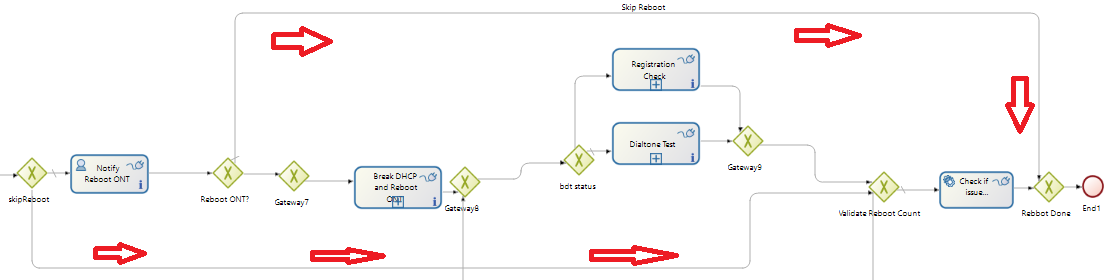
Operations on the node are to be assigned after the completion of the activity as shown below:



After completion of this Activities on the node, The flow execution resumes and will reach **Gateway8** and thereby will reach to **bdt status** (XOR Gateway) to perform operations again. This loop of revalidating flow ends once it reaches the counter in the condition specified on connector from **ValidateRebootCount** gateway.



Also Finally there is another scenario, where flow could be ended when certain conditions on the connectors from gateways **skipReboot** and **Reboot ONT?** are met.



**Finally here is the BOS file that needs to be imported in Bonitasoft for your quick reference:**

****