Use of Action 'Queuing' in SAP MII

Applies to:

SAP MII 11.5 or higher

For more information, visit the Manufacturing homepage.

Summary

This article will provide some tricky and trials on usage of rarely-used action block named 'Queuing' in MII

Author: Som Sarkar

Company: IBM India Pvt Ltd

Created on: 10 November 2008

Author Bio

Som Sarkar is currently working as Technical Consultant in SAP MII. He has nearly 3 years experience in software development

Table of Contents

Introduction

Since long days there is no help documentation or step-by-step guidelines on Queuing Action in MII 12.0.I wonder if I could try my hands to make a TRIAL-N-PLAY to understand the utility of Queuing and look into "How to use Queuing". After a few trials, I finally succeeded to establish the usages and guidelines on Queuing Action for MII developers who want to work or test and apply the concepts of Queue in that action.

Step-By-Steps

Here follows the instructions on Use of Queuing with a small Application on "Managing Employee Data"

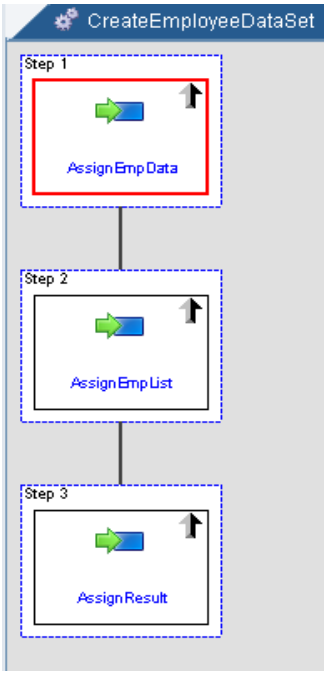
Business Logic Services

Step 1

To develop this Application we need to create two Business Logic Services (BLS) – one for creating Employee Data List (First Name and Last Name), another for managing on Employee Data

Step 2

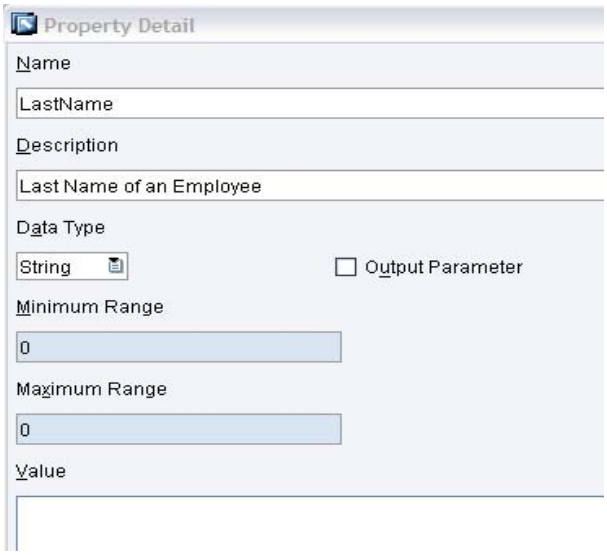
Let me show you the screenshot of first BLS (BLS_CreateEmployeeData) as shown below:



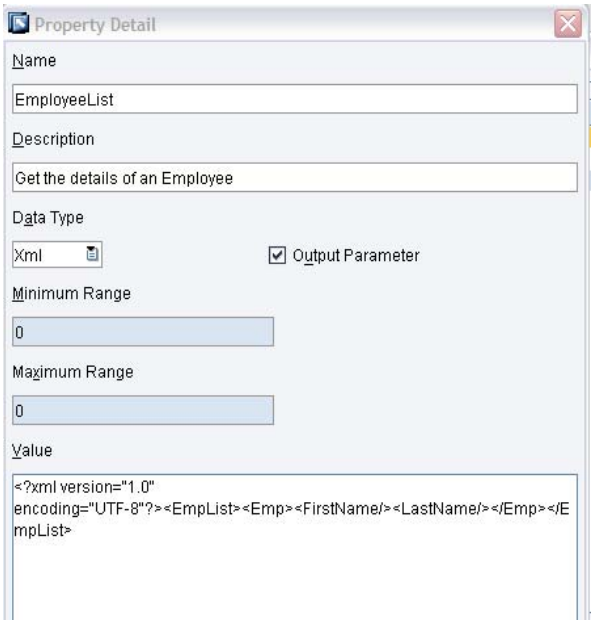
A. Define 3 Transaction Properties as follows

FirstName

LastName

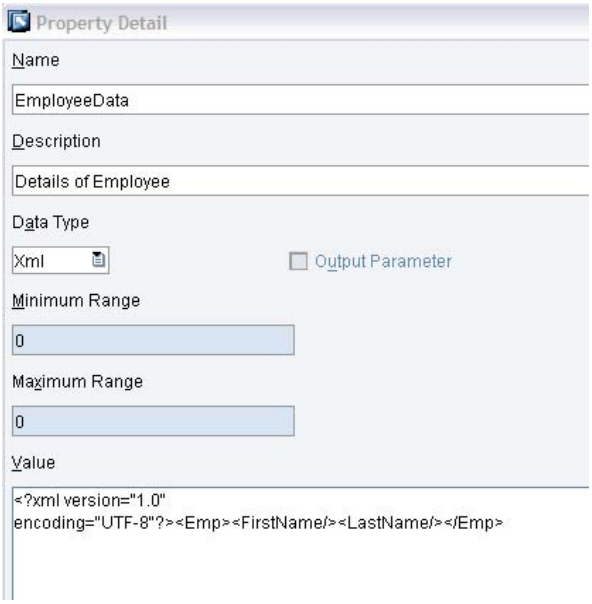


EmployeeList



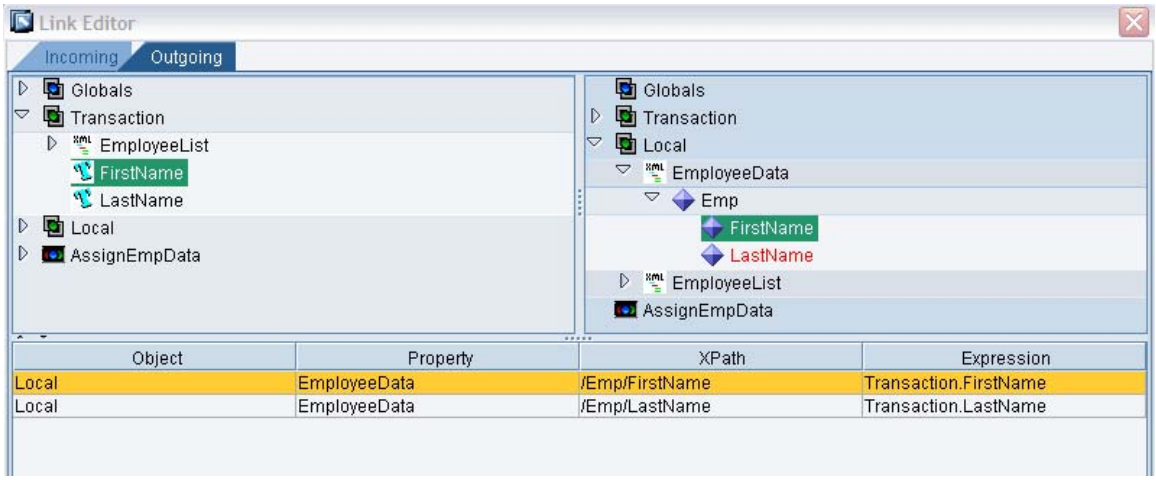
B. Define 2 local variables as follows

EmployeeData

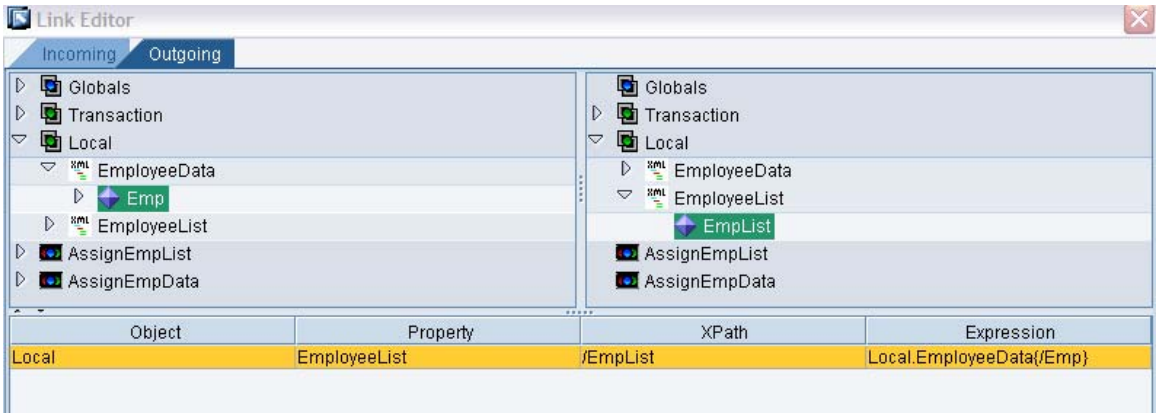


EmployeeList

C. Put Assignment Action in the first sequence and map the transaction variables FirstName and Last Name to the nodes FirstName and LastName defined in the Local variable EmployeeData as follows

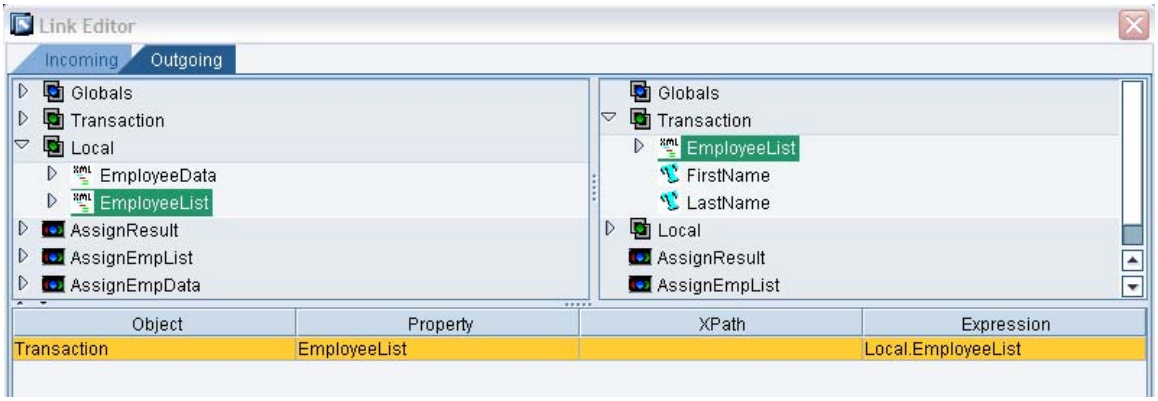


D. Put Assignment Action in the second sequence and map the node Emp defined in the local Variable EmployeeData to the node EmpList defined in the local variable EmployeeList as follows :



Remember to set the Link Type as '**Append XML**'

E. Put Assingment Action in the 3rd (last) sequence and map the local variable EmployeeList to the Transaction Variable EmployeeList as follows:



Step 3

Thus we have created a BLS for creating a Data Set for an Employee which will return the Data in the form of XM. Now we will create another BLS for managing on Employee Data which is a combination of all action blocks of Queuing and other some action blocks

Step 4

See the screen shot of BLS (BLS_ManageEmpData) as below :

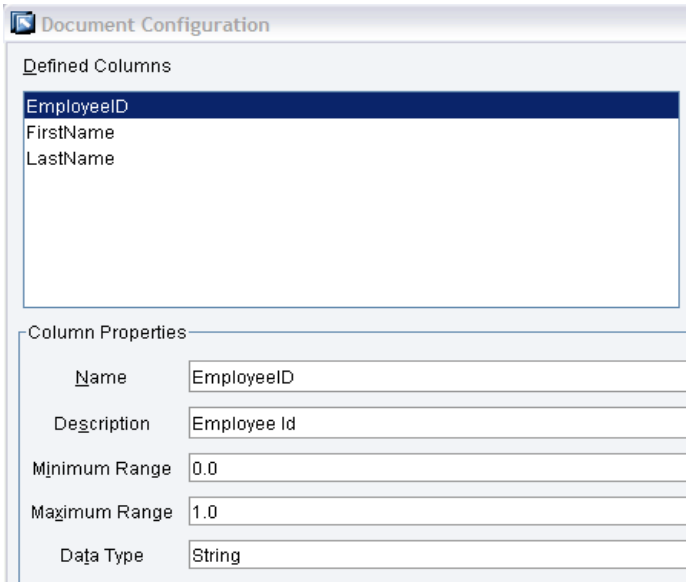
Screen Shot Image

A. Define 6 Transaction Properties : EmpID (String, Input), FirstName (String, Input), LastName (String, Input), QueueFlag (String, Input), Message (String, Output) and GetEmployeeData (XML, Output)

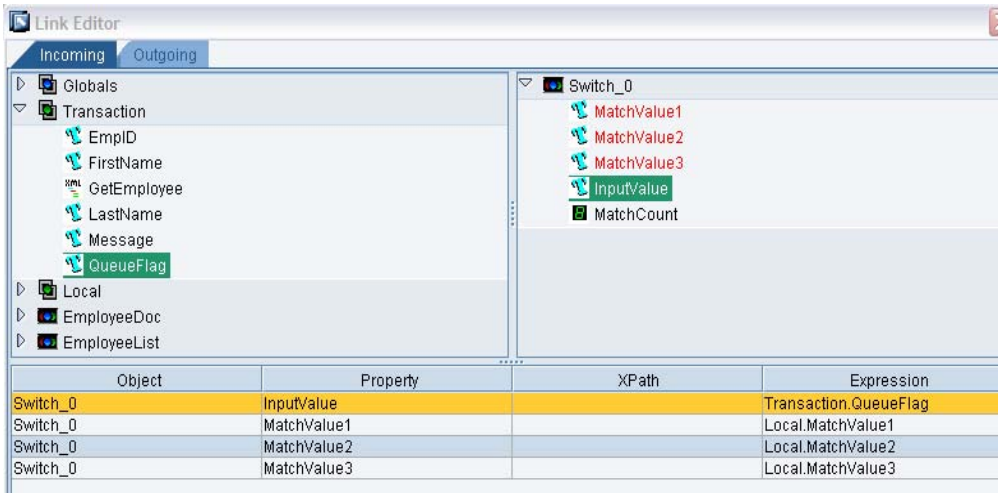
B. Define local variables for 3 MatchValues used for Switch Action : MatchValue1 – PQ (for Queue Put), MatchValue2 – GQ (for Queue Get) and MatchValue3 – DQ (for Queue Delete)

C. Put the Action 'Queue List' in the first sequence and configure as follows

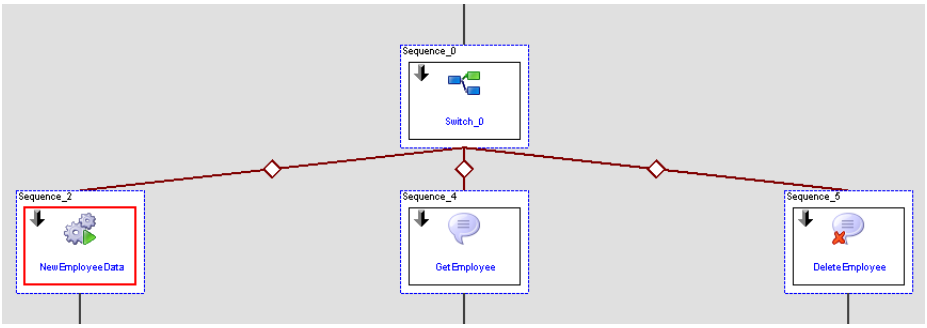
D. Put the Action 'Document' in the 2nd sequence and define 3 columns as follows with all data types as String



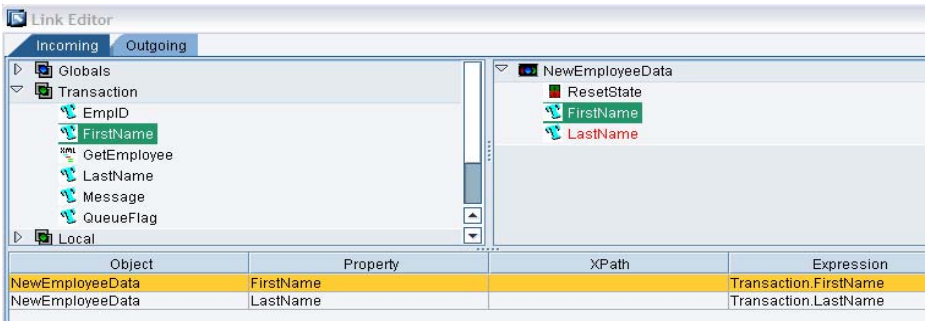
E. Put the Action 'Switch' in the 3rd sequence and set number of inputs as 3. Configure the links between transaction properties and Switch properties as follows



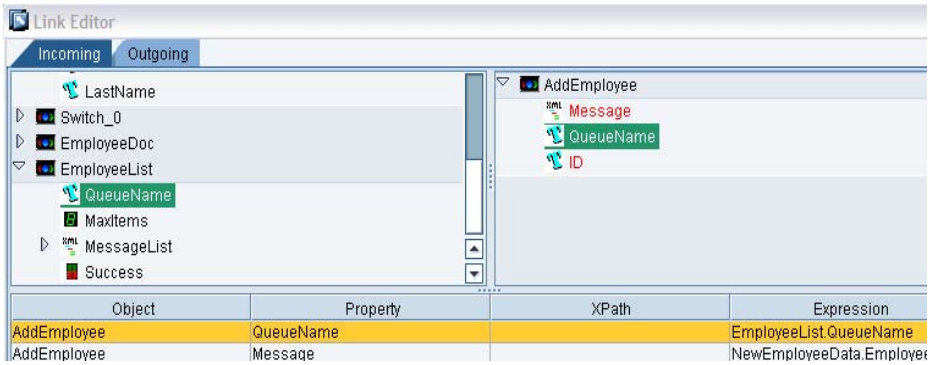
F. Add 3 sequences under the sequence defined in C. and put the Actions Transaction Call, Queue Get and Queue Delete respectively as shown below:



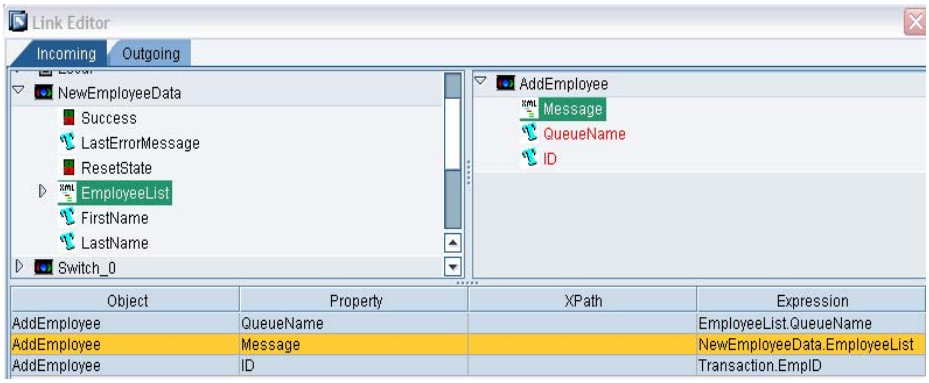
G. In Action 'Transaction Call', call the transaction which was created in the previous BLS (BLS_CreateEmployeeData) and set the links as follows :



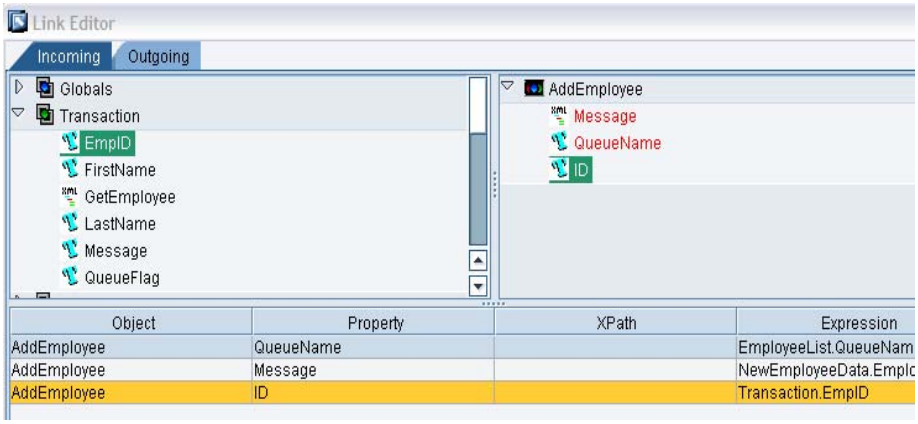
H. In the next sequence to sequence of Transaction Call, put the Action 'Queue Put'. And set the links of all 3 nodes of the action as follows



Map the node QueueName of the action 'Queue List' (in the left panel) defined in the 1st sequence to the node QueueName of the action 'Queue Put' (in the right panel)



Map the node EmployeeList of the action 'Transaction Call' (in the left panel) defined in the previous sequence to the node Message of the action 'Queue Put' (in the right panel)



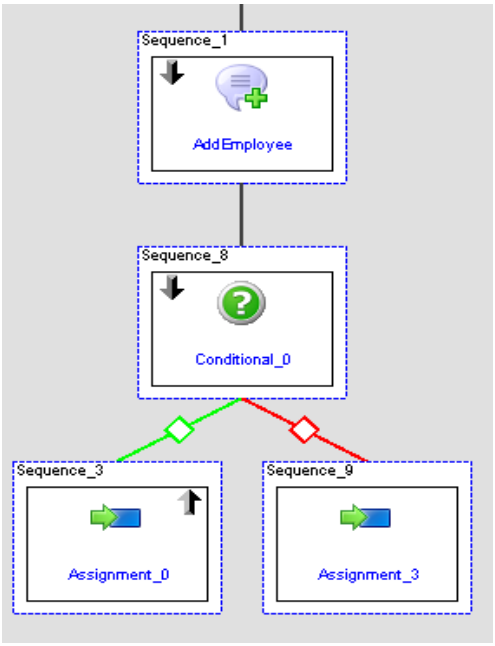
Map the node EmpID of the Transaction (in the left panel) to the node ID of the action 'Queue Put' (in the right panel)

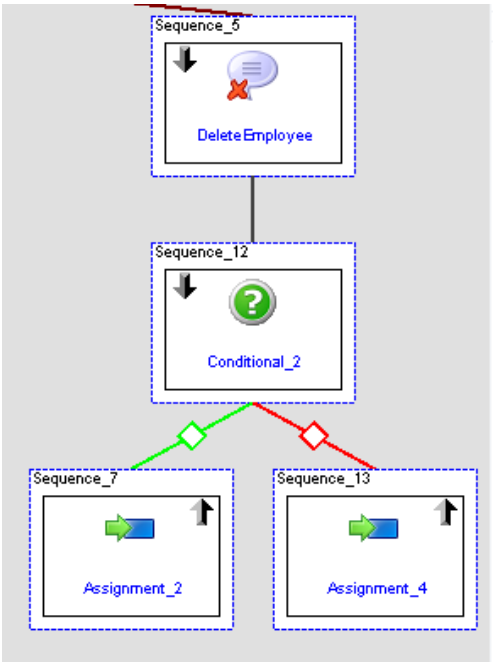
I. In the action 'Queue Get' do the followings

- Map the node EmpID of the Transaction (in the left panel) to the node ID of the action 'Queue Get' (in the right panel)
- Map the node QueueName of the action 'Queue List ' (in the left panel) to the node QueueName of the action 'Queue Get' (in the right panel)

J. Similarly do the same needful things for the action 'Queue Delete'

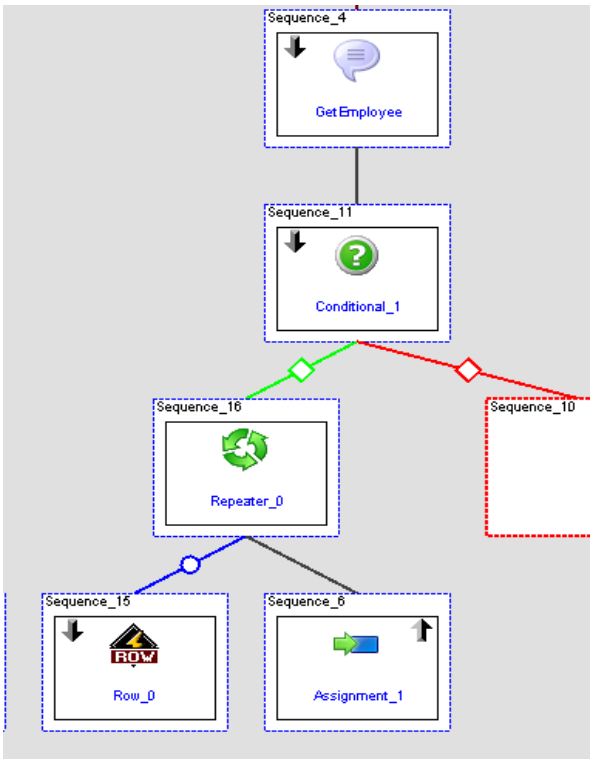
K. In the next sequence to the sequence of 'Queue Put' and 'Queue Delete'put remaining actions as follows





- The action 'Condition' will check the success of the operation of 'Queue Put'/'Queue Delete'
- On success of the operation, Assign the local variable (PQ_Message/DQ_Message) to the Transaction variable (Message)
- On Failure of the operation, similarly assign the local variable (PQ_ErrorMsg/DQ_Message) to the Transaction variable (Message)

L. In the next sequence of 'Queue Get' put the actions as follows :



- Check the success of the action 'Queue Get'
- On Success of the action, loop the Message (XML node) of the 'Queue Get' with XPath Expression GetEmployee.Message{/EmpList/Emp}
- Then append the employee data (First Name and Last Name) from the message loop and ID node of 'Queue Get' to the corresponding columns defined in the XML Document created in previous step (B)
- Finally assign the XML document output to the transaction output variable.

Step 5

Thus we have created the BLS which will manage the Employee Data with three operations : Add employee (Using 'Queue Put'), Get Employee (Using 'Queue Get') and Delete employee (Using 'Queue Delete')

Step 6

Now we can test the use of Queuing for all three functionalities with the help of Xacute Query as follows

1. Add Employee :
    - In an Xacute Query, select the Transaction we have created previously.
    - Set the Output Parameter with 'Message' from the list of available output parameters of the selected transaction
    - Assign the input parameters as shown below
    - Enter values for Employee Id, FirstName and Last Name
    - Enter 'PQ' for the Flag for Queuing operation (for the action 'Queue Put')
    - Execute the Xacute Query
2. Delete Employee :
    - In an Xacute Query, select the Transaction we have created previously.
    - Set the Output Parameter with 'Message' from the list of available output parameters of the selected transaction
    - Assign the input parameters as shown below
    - Enter values for Employee Id and leave the input parameters for FirstName and Last Name as blank
    - Enter 'DQ' for the Flag for Queuing operation (for the action 'Queue Put')
    - Execute the Xacute Query

3. Get Employee :

    - In an Xacute Query, select the Transaction we have created previously.
    - Set the Output Parameter with 'GetEmployeeData' from the list of available output parameters of the selected transaction
    - Assign the input parameters as shown below
    - Enter values for Employee Id and leave the input parameters for FirstName and Last Name as blank
    - Enter 'GQ' for the Flag for Queuing operation (for the action 'Queue Put')
    - Execute the Xacute Query

Here, we are now able to use Queuing Action in MII 12.0.

Related Content

SAP MII12.0 Help

For more information, visit the Manufacturing homepage.

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.