Adeptia SAP BAPI Connector

Adeptia Suite

Overview

- Setup SAP BAPI Connector
  - Install SAP BAPI Connector
  - Modify “Server configuration.properties” file
  - Configure SAP configuration

- Using SAP BAPI Connector
  - Inbuilt sample process flow SAP to Excel File
  - Create new process flow as per your requirement
  - Using Multiple SAP functions in a Single Process Flow.

Install SAP BAPI Connector

- Pre-requisite:
  - Adeptia Suite must be installed on Windows Server 2008 X64 OS

  Note: Adeptia SAP BAPI connector has bundled JCO (Jar & DLL) for 64 bit windows. To use connecter with 32 bit or other OS, you need to use respective JCO from your SAP install. Refer Appendix .

- Steps to install SAP BAPI connector:
  - Extract the SAPDemo.zip in a folder.
  - From extracted/sap folder, copy “sapjco3.dll” and paste it in “C:\Windows\System32”.
  - SAP accelerator will be installed automatically in the environment when you will deploy the zip.

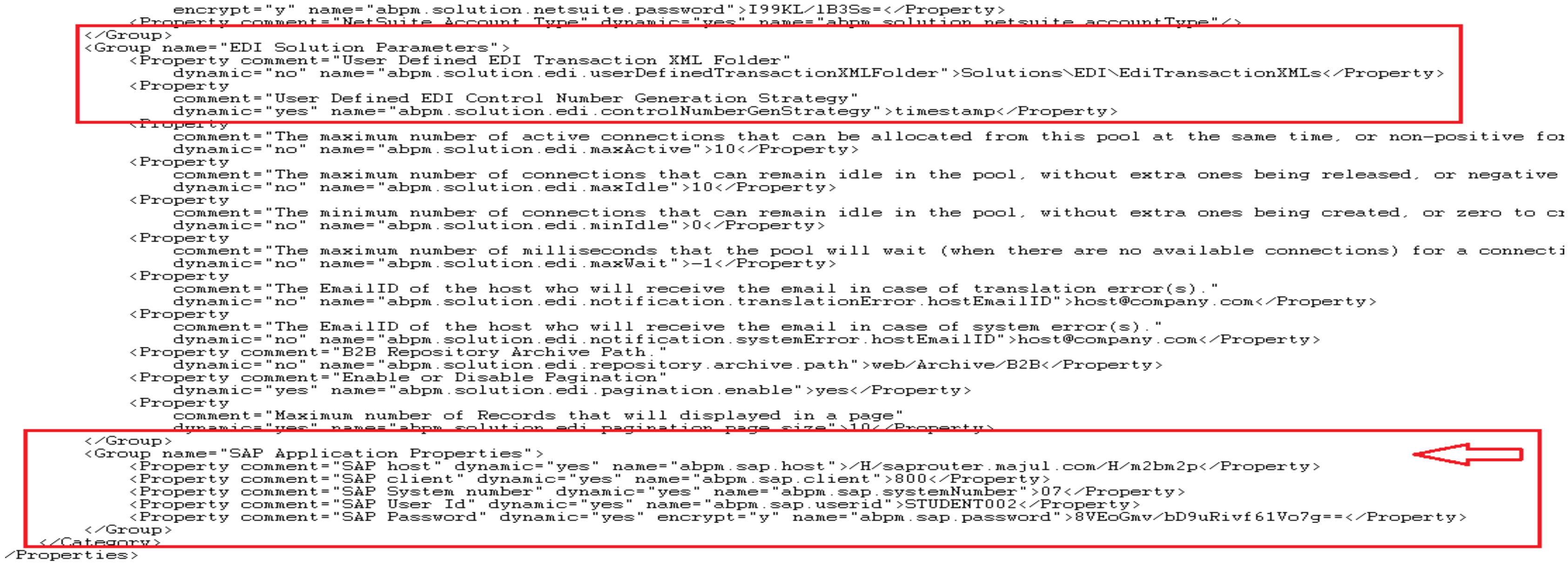
Modify “Server-configuration.properties”

To add SAP server configuration properties:

- Open Server-Configure.Properties file from <AdeptiaInstallFolder>/ServerKernel/etc folder and copy the content displayed from below box:

```
<Group name="SAP Application Properties">
        <Property comment="SAP host" dynamic="yes" name="abpm.sap.host"></Property>
        <Property comment="SAP client" dynamic="yes" name="abpm.sap.client"></Property>
        <Property comment="SAP System number" dynamic="yes" name="abpm.sap.systemNumber"></Property>
        <Property comment="SAP User Id" dynamic="yes" name="abpm.sap.userid"></Property>
        <Property comment="SAP Password" dynamic="yes" encrypt="y" name="abpm.sap.password"></Property>
     </Group>
```

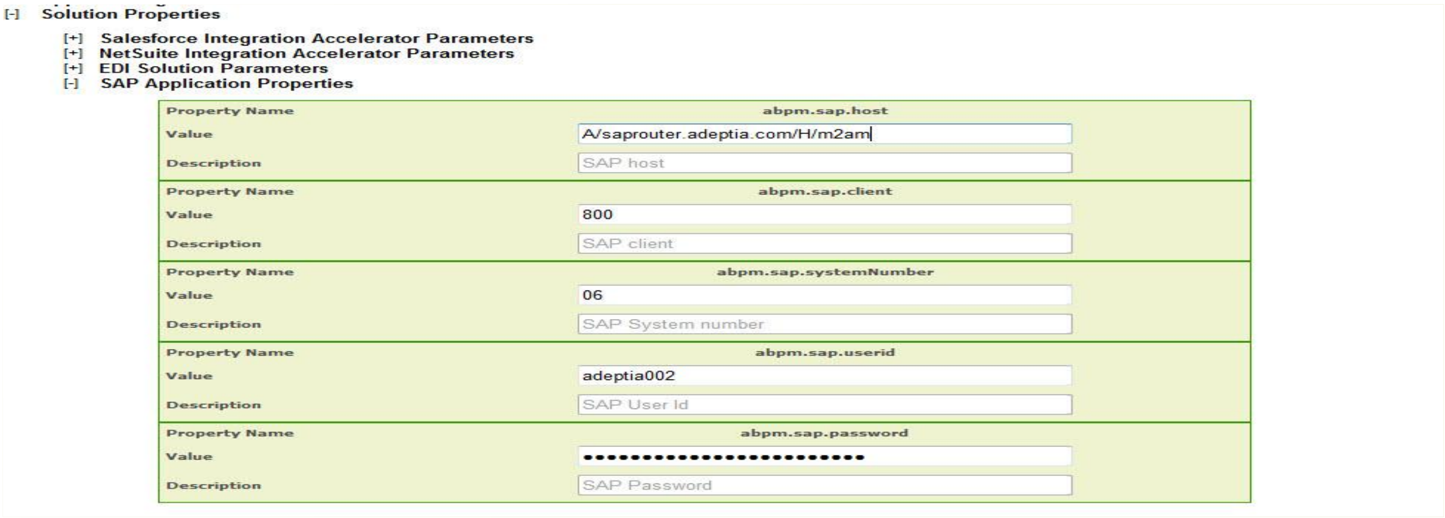
- Copy the above content on the position as shown in below screenshot.



```
            encrypt="y" name="abpm.solution.netsuite.password">I99KL/1B3Ss=</Property>
        <Property comment="NetSuite Account Type" dynamic="yes" name="abpm.solution.netsuite.accountType"/>
</Group>
<Group name="EDI Solution Parameters">
        <Property comment="User Defined EDI Transaction XML Folder"
            dynamic="no" name="abpm.solution.edi.userDefinedTransactionXMLFolder">Solutions\EDI\EdiTransactionXMLs</Property>
        <Property
            comment="User Defined EDI Control Number Generation Strategy"
            dynamic="yes" name="abpm.solution.edi.controlNumberGenStrategy">timestamp</Property>
        <Property
            comment="The maximum number of active connections that can be allocated from this pool at the same time, or non-positive for
            dynamic="no" name="abpm.solution.edi.maxActive">10</Property>
        <Property
            comment="The maximum number of connections that can remain idle in the pool, without extra ones being released, or negative
            dynamic="no" name="abpm.solution.edi.maxIdle">10</Property>
        <Property
            comment="The minimum number of connections that can remain idle in the pool, without extra ones being created, or zero to cr
            dynamic="no" name="abpm.solution.edi.minIdle">0</Property>
        <Property
            comment="The maximum number of milliseconds that the pool will wait (when there are no available connections) for a connecti
            dynamic="no" name="abpm.solution.edi.maxWait">-1</Property>
        <Property
            comment="The EmailID of the host who will receive the email in case of translation error(s)."
            dynamic="no" name="abpm.solution.edi.notification.translationError.hostEmailID">host@company.com</Property>
        <Property
            comment="The EmailID of the host who will receive the email in case of system error(s)."
            dynamic="no" name="abpm.solution.edi.notification.systemError.hostEmailID">host@company.com</Property>
        <Property comment="B2B Repository Archive Path."
            dynamic="no" name="abpm.solution.edi.repository.archive.path">web/Archive/B2B</Property>
        <Property comment="Enable or Disable Pagination"
            dynamic="yes" name="abpm.solution.edi.pagination.enable">yes</Property>
        <Property
            comment="Maximum number of Records that will displayed in a page"
            dynamic="yes" name="abpm.solution.edi.pagination.page.size">10</Property>
    </Group>
    <Group name="SAP Application Properties">
        <Property comment="SAP host" dynamic="yes" name="abpm.sap.host">/H/saprouter.majul.com/H/m2bm2p</Property>
        <Property comment="SAP client" dynamic="yes" name="abpm.sap.client">800</Property>
        <Property comment="SAP System number" dynamic="yes" name="abpm.sap.systemNumber">07</Property>
        <Property comment="SAP User Id" dynamic="yes" name="abpm.sap.userid">STUDENT002</Property>
        <Property comment="SAP Password" dynamic="yes" encrypt="y" name="abpm.sap.password">8VEoGmv/bD9uRivf61Vo7g==</Property>
    </Group>
  </Category>
/Properties>
```

Add SAP Configuration

- Start the Kernel and WebRunner.
- Login as admin, Go to “Administer > Setup > Application Settings > Update system properties > SolutionProperties > SAP ApplicationProperties” .
- Configure your SAP parameters as shown in below screen. This is a sample configuration, you need to add the parameters as per you SAP install.



[-] Solution Properties
- [+] Salesforce Integration Accelerator Parameters
- [+] NetSuite Integration Accelerator Parameters
- [+] EDI Solution Parameters
- [-] SAP Application Properties

Property Name	abpm.sap.host
Value	A/saprouter.adeptia.com/H/m2am
Description	SAP host
Property Name	abpm.sap.client
Value	800
Description	SAP client
Property Name	abpm.sap.systemNumber
Value	06
Description	SAP System number
Property Name	abpm.sap.userid
Value	adeptia002
Description	SAP User Id
Property Name	abpm.sap.password
Value	●●●●●●●●●●●●●●●●●●●●●●●●●
Description	SAP Password

- Re-start the Adeptia Suite Kernel and WebRunner.

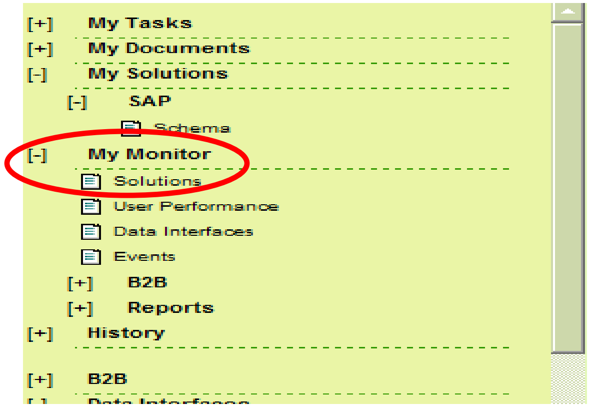
Inbuilt Sample Process flow (SAP to Database) for Function BAPI_PO_GETITEMSREL

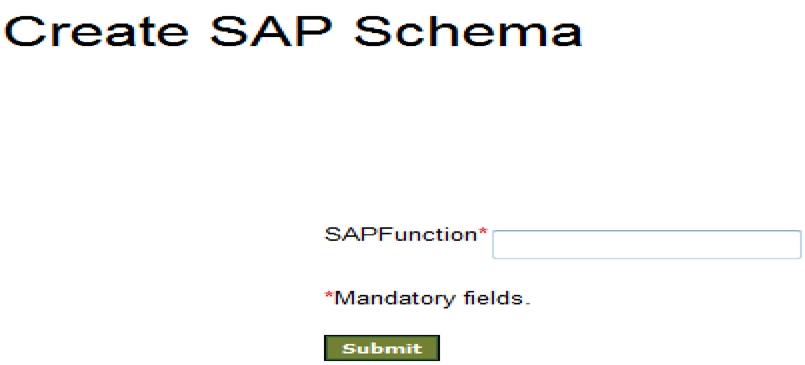
- Login as
  - UserID: SAPDemoUser
  - Password: sapdemouser123
- Modify the target path
  - Edit the File Target activity “POrder_Source” and modify the File path as per your need.
- Process Flow
  - Name: 'PO_SAP_To_Excel**'**
  - Description**:** Synch SAP PO with staging Excel file
  - This Process calls SAP “BAPI_PO_GETITEMSREL” function.
- Execute the process flow.
  - Go to *Design* > *Process Flow* in Adeptia Suite.
  - Select *PO_SAP_To_Excel* process flow and execute.
- Input
  - File Source Name: 'POrder_Source**'**
  - File path: .\Solutions\Demo\input\PurchaseOrder.xml

  Note: This file has SAP request to fetch “Purchase Order”
- Output
  - An excel will be created in the folder specified in File Target “SAPTarget”.

Create New Process flow for another SAP function

- Create Input and output schema activity for new SAP function, using Interface below.
- Enter the SAP function name such as “BAPI_PO_GETDETAIL” and click Submit. It will create BAPI_PO_GETDETAIL_Input_Schema and BAPI_PO_GETDETAIL_Output_Schema.

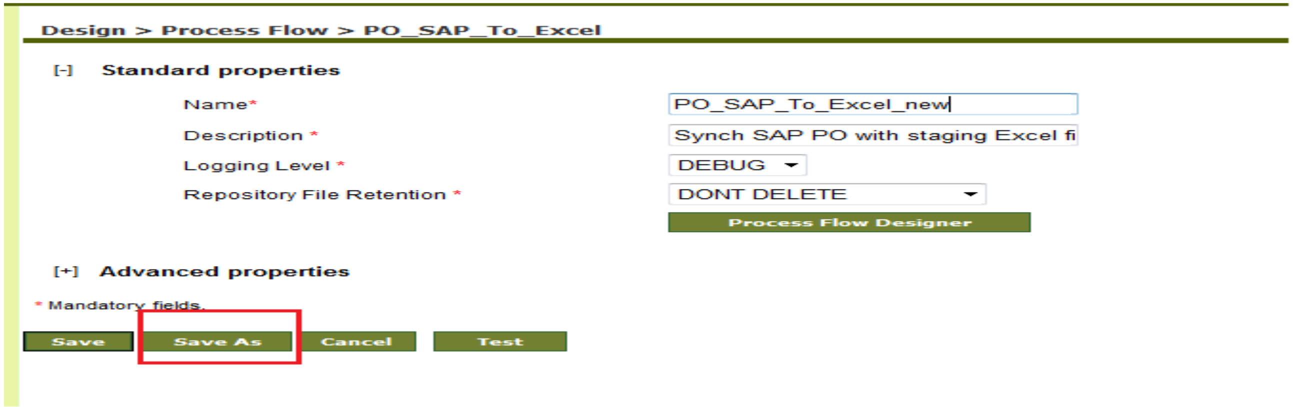




- The input XML Schema is used to give the input to SAP function.
- Create an XML file as per input XML schema. The Input XML schema name will be “SAPFunction”_Input_Schema for example (BAPI_PO_GETDETAIL_Input_Schema).
- Create a FileSource activity to pick this input file.
- Create a Target activity and the schema as per your requirement.

Create New Process flow for Another SAP function

- Create a mapping activity and map output Schema to Target Schema. The Output Schema name will be “SAPFunction”_Output_Schema for example (BAPI_PO_GETDETAIL_Output_Schema)
- Open the process flow(PO_SAP_To_Excel) in edit mode, change its name as per your need and click “Save As”. As shown below:



Design > Process Flow > PO_SAP_To_Excel

[-] Standard properties

Name* PO_SAP_To_Excel_new

Description * Synch SAP PO with staging Excel fi

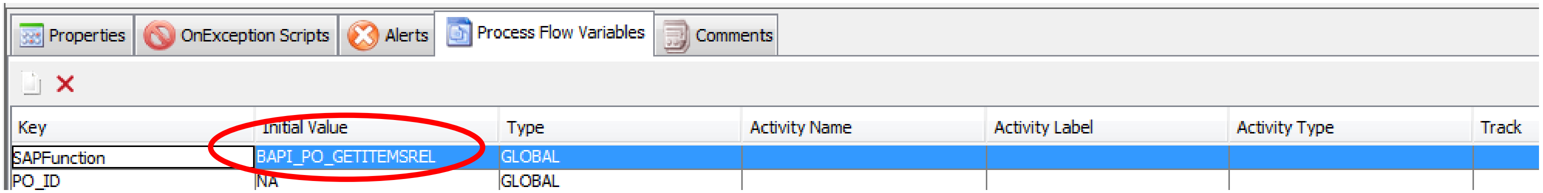
Logging Level * DEBUG

Repository File Retention * DONT DELETE

[+] Advanced properties

* Mandatory fields.

- Now edit this New process flow and attach the above created Source, Mapping & Target activities.
- Click on Process Flow Variable and enter the name of the SAP function which you want to call in the Initial Value textbox as shown below.



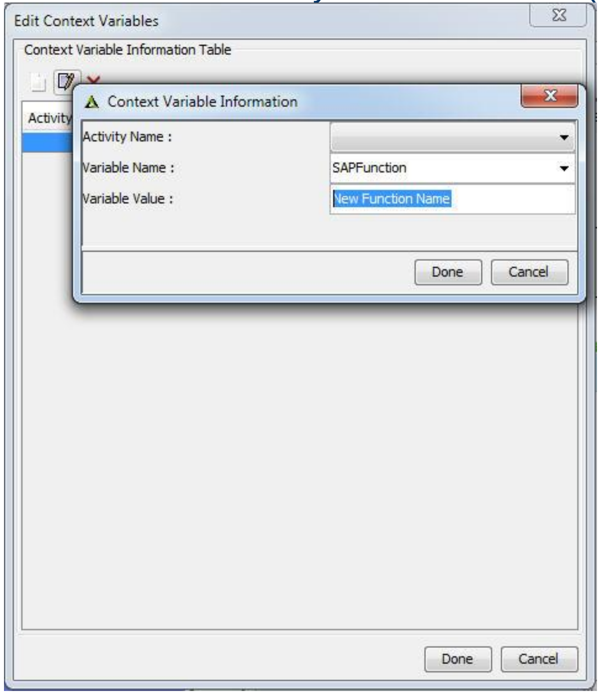
Key	Initial Value	Type	Activity Name	Activity Label	Activity Type	Track
SAPFunction	BAPI_PO_GETITEMSREL	GLOBAL				
PO_ID	NA	GLOBAL				

- Save the process flow and execute it.

Using Multiple SAP functions in a Single Process Flow.

You can invoke multiple SAP function within single process flow. To invoke multiple SAP function, you need to modify the process flow as per the steps given below.

- Create Input and output schema activity for new SAP function.
- Open the process flow in edit mode.
- Use put-context-var activity at the end of the process flow.
- Double click this put-context-var, and click on Edit Context Variable in bottom after that select *SAPFunction* in the **Variable Name** dropdown field**.**
- In the **Variable Value** field , enter the name of the new SAP function that you want to invoke (Refer to the below Screenshot)

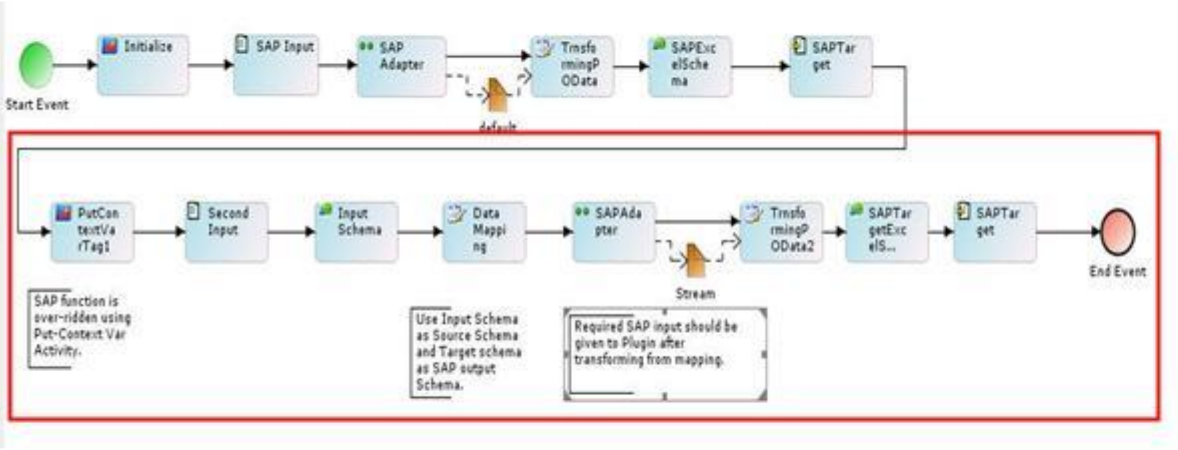

Edit Context Variables
Context Variable Information Table
Activity
Context Variable Information
Done
Cancel
Done
Cancel

Activity Name :

Variable Name : SAPFunction

Variable Value : New Function Name

- After that use the source activity that will provide Input for SAP function.
- Use the Schema activity as per the format of the source file. For example if the source file is in text format, use the Text Schema.
- Create a mapping activity to transform the input in the format that is to be given to SAP Adapter. In this mapping, at source end use the schema activity that you have created as per the source file. At the target, you need to use the input Schema of the new SAP function.
- You need to give the output of this mapping activity to the SAP Adapter custom plugin.
- After the SAP adapter custom plugin, you can directly use the target activity to write the output or you can transform the output into required format.
- To convert the output in the required format, you need to use the mapping activity as shown in the figure below.



SAP function is over-ridden using Put-Context Var Activity.

Use Input Schema as Source Schema and Target schema as SAP output Schema.

Required SAP input should be given to Plugin after transforming from mapping.

- In this mapping activity, at source end you have to use output schema of the new SAP function.
- In the target side of the this mapping you need to use the schema as per the format of the target data.
- Similarly, you can use other SAP functions in a single flow by repeating above steps.

Appendix: Copying SAP jar and dll for other OS

- For 32 bit Windows Platform:
  - For 32 bit windows machine, it needs respective JCO from SAP install and and copy it in the folder “web/libs/sap”.