

Lab 1 - SAP Connector (IDoc Inbound Endpoint)

Ver. 1.0.0

Overview

This lab will show you how to use the SAP Connector as an IDoc Inbound Endpoint within your flow. The project, once deployed, will serve as a JCo Server and listen for an IDoc of type DEBMAS. Once it receives the IDoc, it will output the content to a file on your local machine.

Reminder: This lab requires the use of your own SAP instance. If you don't have one, you can leverage a hosted SAP service. For the purposes of this lab, we utilized [Sandbox SAP](#).

[Overview](#)

[Steps](#)

[1. Setup Dependencies](#)

[2. Configure Global Properties for the SAP Connector](#)

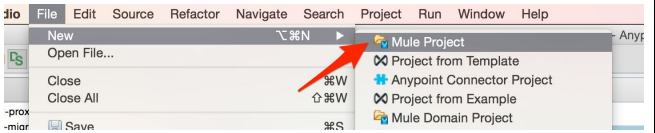
[3. Configure the SAP Connector to receive IDocs](#)

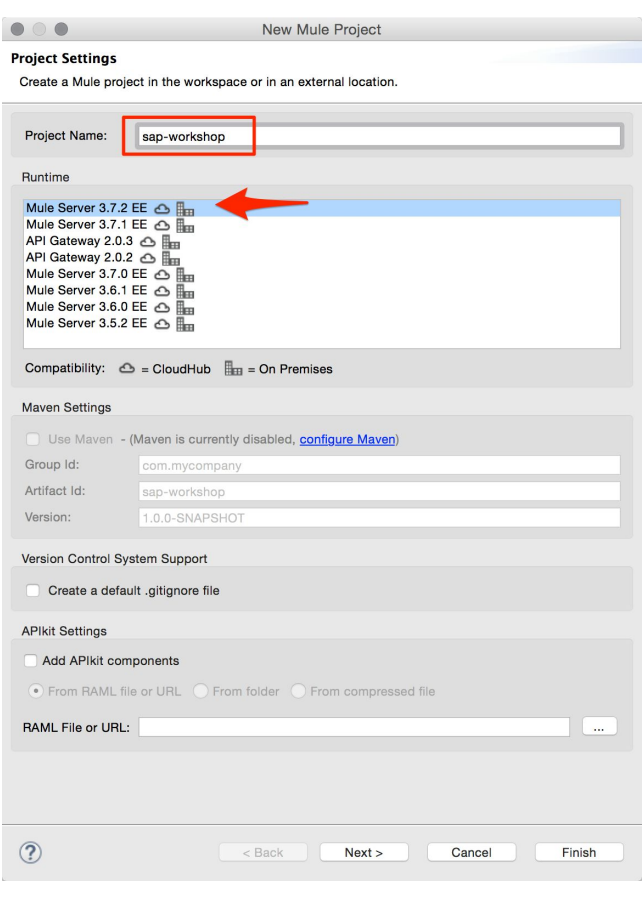
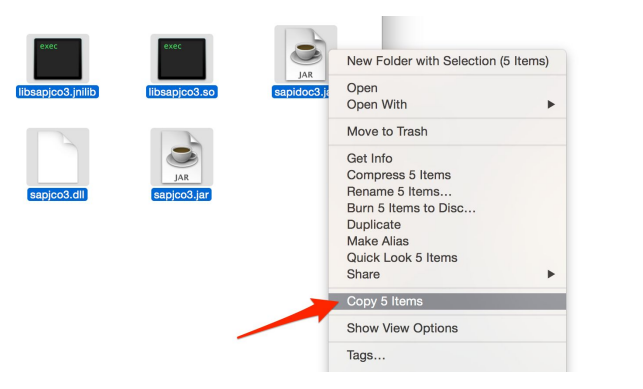
[4. Test SAP IDoc Inbound Endpoint](#)

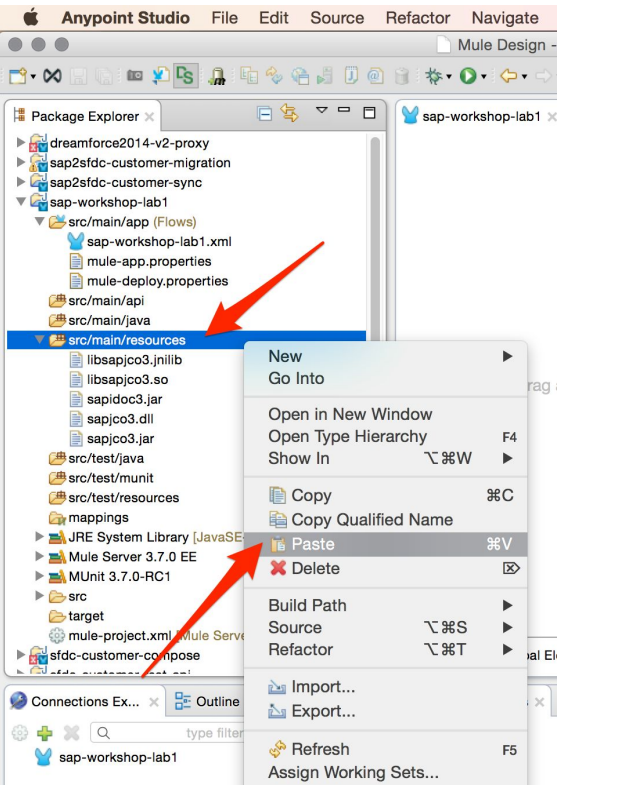
[Summary](#)

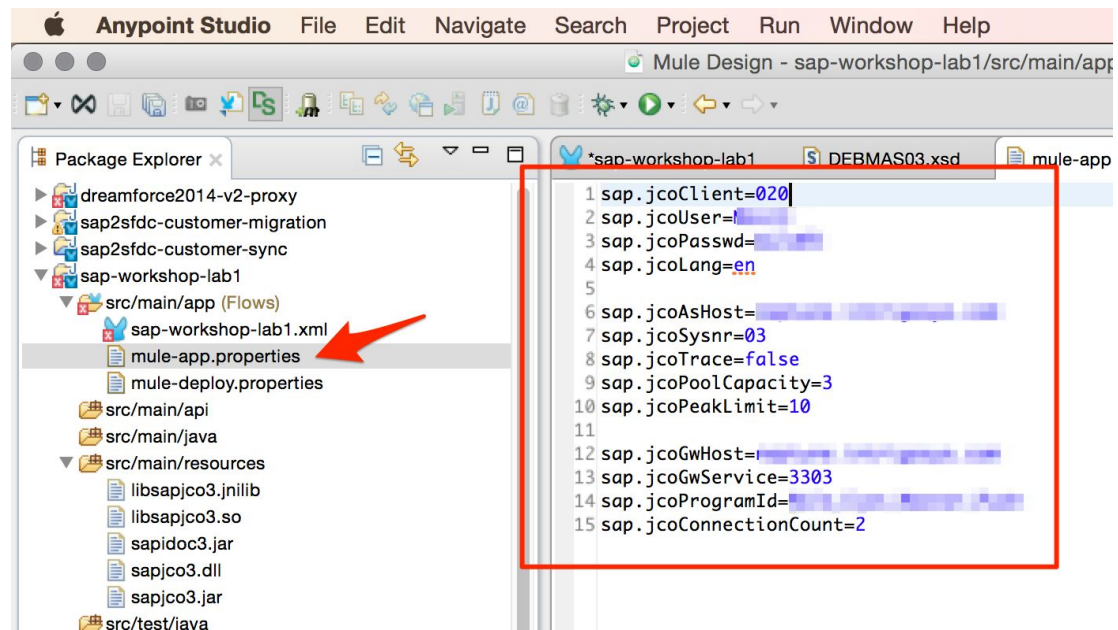
Steps

1. Setup Dependencies

1.1	<p>Open Anypoint Studio and create a new Mule Project.</p> <p>Click on File > New > Mule Project</p>	 A screenshot of the Anypoint Studio application window. The 'File' menu is open, and the 'New' option is selected, which has opened a sub-menu. In this sub-menu, the 'Mule Project' option is highlighted with a red arrow. Other visible options in the sub-menu include 'Project from Template', 'Anypoint Connector Project', 'Project from Example', and 'Mule Domain Project'. The main menu bar at the top includes 'File', 'Edit', 'Source', 'Refactor', 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'.
-----	---	---

<p>1.2</p>	<p>In the New Mule Project dialog window, give your project a Project Name:</p> <p>e.g. sap-workshop</p> <p>Under the Runtime section, select the most recent version of the Mule Server runtime.</p> <p>Click on Finish</p>	
<p>1.3</p>	<p>Before we can use the SAP Connector in our project, we need to add the required SAP library files to our project.</p> <p>In your File Explorer, depending on your OS, select all the necessary files and click on Copy</p>	

1.4	<p>Switch back to Anypoint Studio and locate the src/main/resources folder in your new Mule project within the Package Explorer.</p> <p>Right-click on the folder and select Paste to copy the SAP library files to the folder.</p> <p>You should have the following files added to the folder.</p> <p>Java Connector (JCo)</p> <ul style="list-style-type: none"> ● sapjco3.jar <p>IDoc Library</p> <ul style="list-style-type: none"> ● sapidoc3.jar <p>JCo Platform-specific native libraries:</p> <ul style="list-style-type: none"> ● Windows - sapjco3.dll ● Mac OS - libsapjco3.jnilib ● Linux - libsapjco3.so 																																											
1.5	<p>Next we want to setup property placeholders to store the connection information to your SAP instance.</p> <p>In your Mule project in the Package Explorer, locate the mule-app.properties file under the src/main/app folder.</p> <p>Open that file and add the following lines to the file with your connection strings to your SAP instance. Replace the items in brackets "<>" with the connection settings for your SAP instance.</p> <table border="1" data-bbox="289 1276 1404 1833"> <thead> <tr> <th>Property Placeholder</th><th>Value</th><th>Notes</th></tr> </thead> <tbody> <tr> <td>sap.jcoClient</td><td><Client ID></td><td></td></tr> <tr> <td>sap.jcoUser</td><td><SAP Username></td><td>Username</td></tr> <tr> <td>sap.jcoPasswd</td><td><SAP Password></td><td>Password</td></tr> <tr> <td>sap.jcoLang</td><td>EN</td><td></td></tr> <tr> <td>sap.jcoAsHost</td><td><Application Server></td><td>URL or IP address of SAP system.</td></tr> <tr> <td>sap.jcoSysnr</td><td><Instance Number></td><td>System number used to connect to SAP</td></tr> <tr> <td>sap.jcoTrace</td><td>false</td><td></td></tr> <tr> <td>sap.jcoPoolCapacity</td><td>3</td><td></td></tr> <tr> <td>sap.jcoPeakLimit</td><td>10</td><td></td></tr> <tr> <td>sap.jcoGwHost</td><td><Application Server></td><td>URL or IP address of SAP system.</td></tr> <tr> <td>sap.jcoGwService</td><td><SAP Gateway></td><td>Service Mappings</td></tr> <tr> <td>sap.jcoProgramId</td><td>MULE_LAB</td><td>This field corresponds to the Program ID that you setup in the SAP Setup guide in Step 2.</td></tr> <tr> <td>sap.jcoConnectionCount</td><td>2</td><td></td></tr> </tbody> </table>	Property Placeholder	Value	Notes	sap.jcoClient	<Client ID>		sap.jcoUser	<SAP Username>	Username	sap.jcoPasswd	<SAP Password>	Password	sap.jcoLang	EN		sap.jcoAsHost	<Application Server>	URL or IP address of SAP system.	sap.jcoSysnr	<Instance Number>	System number used to connect to SAP	sap.jcoTrace	false		sap.jcoPoolCapacity	3		sap.jcoPeakLimit	10		sap.jcoGwHost	<Application Server>	URL or IP address of SAP system.	sap.jcoGwService	<SAP Gateway>	Service Mappings	sap.jcoProgramId	MULE_LAB	This field corresponds to the Program ID that you setup in the SAP Setup guide in Step 2.	sap.jcoConnectionCount	2		
Property Placeholder	Value	Notes																																										
sap.jcoClient	<Client ID>																																											
sap.jcoUser	<SAP Username>	Username																																										
sap.jcoPasswd	<SAP Password>	Password																																										
sap.jcoLang	EN																																											
sap.jcoAsHost	<Application Server>	URL or IP address of SAP system.																																										
sap.jcoSysnr	<Instance Number>	System number used to connect to SAP																																										
sap.jcoTrace	false																																											
sap.jcoPoolCapacity	3																																											
sap.jcoPeakLimit	10																																											
sap.jcoGwHost	<Application Server>	URL or IP address of SAP system.																																										
sap.jcoGwService	<SAP Gateway>	Service Mappings																																										
sap.jcoProgramId	MULE_LAB	This field corresponds to the Program ID that you setup in the SAP Setup guide in Step 2.																																										
sap.jcoConnectionCount	2																																											



Additional Information

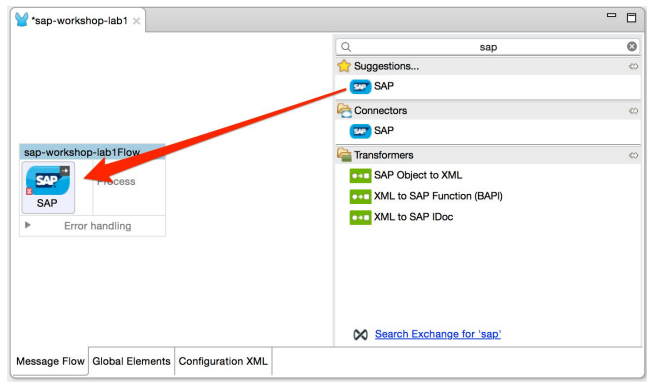
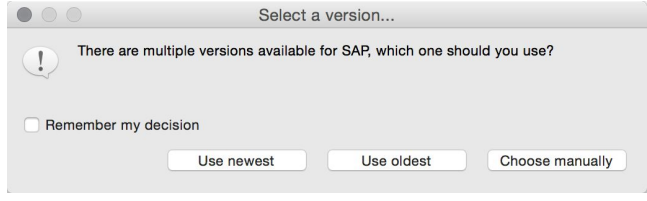

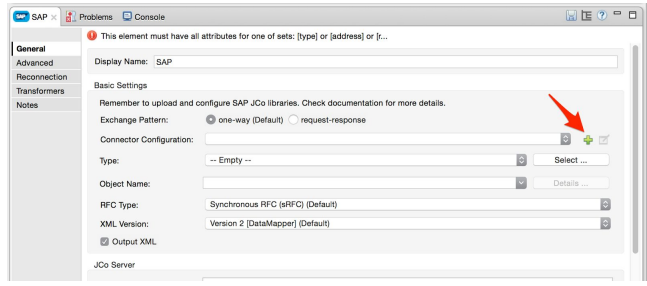
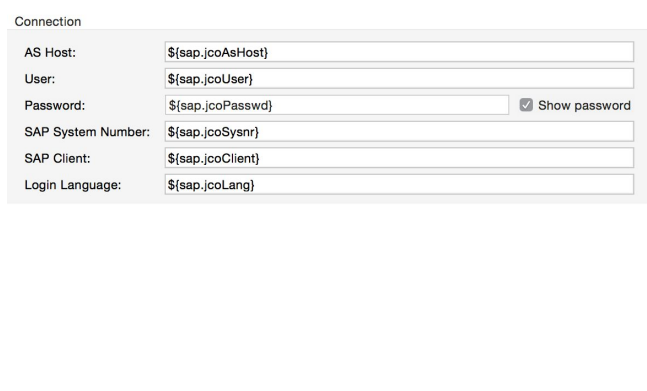
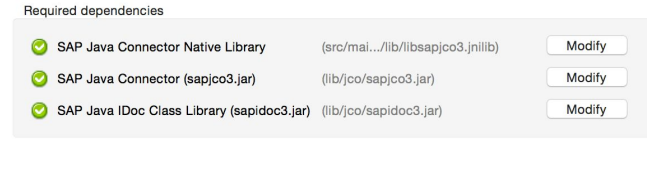
SAP Connector

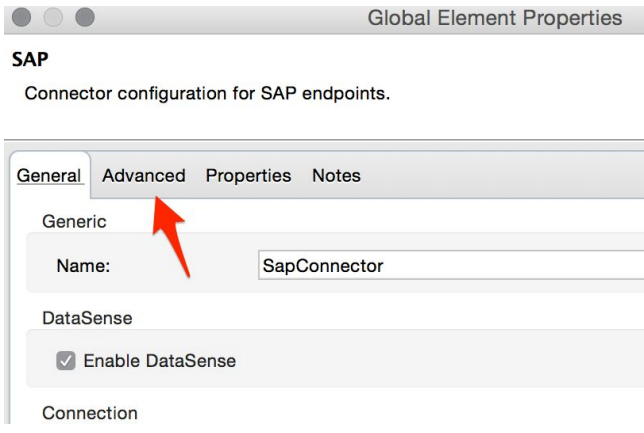
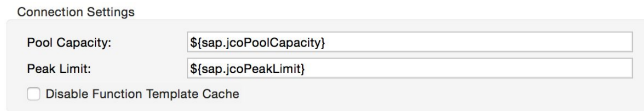
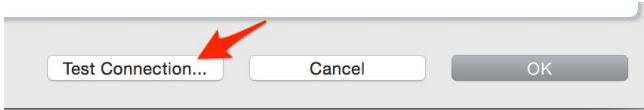
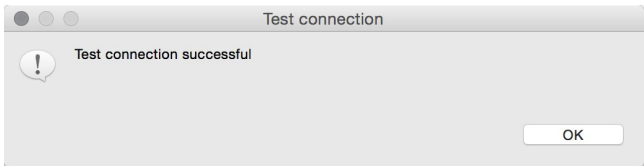
<https://docs.mulesoft.com/mule-user-guide/v/3.7/sap-connector#connector-global-element>

Configuring Properties


<https://developer.mulesoft.com/docs/display/current/Configuring+Properties>

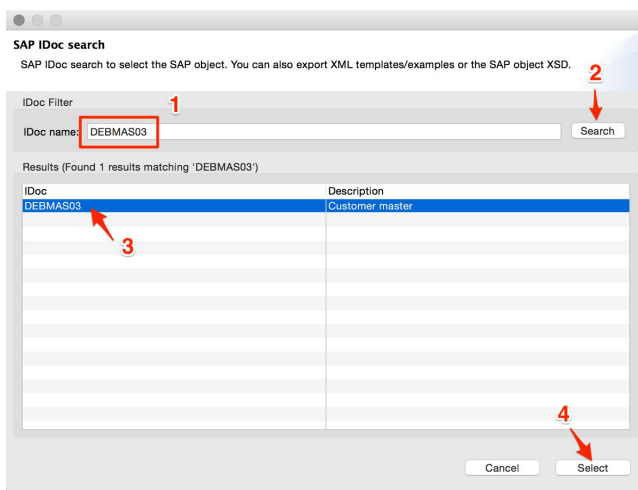
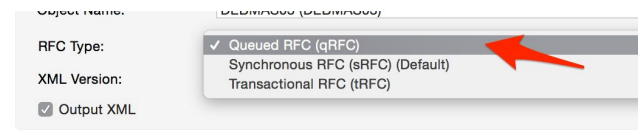

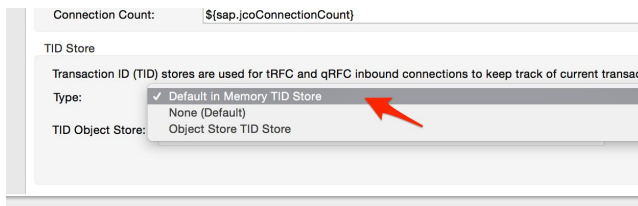
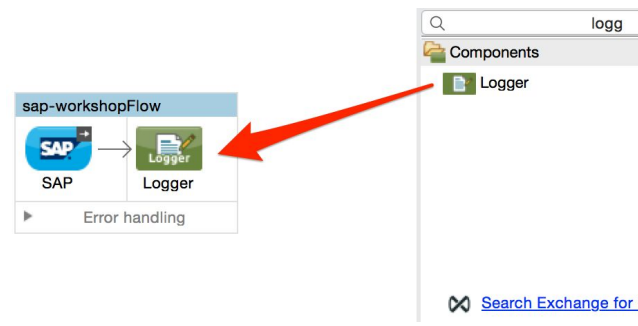
2. Configure Global Properties for the SAP Connector

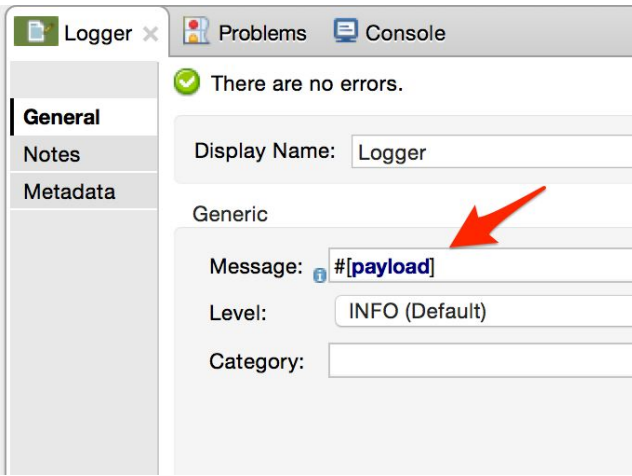
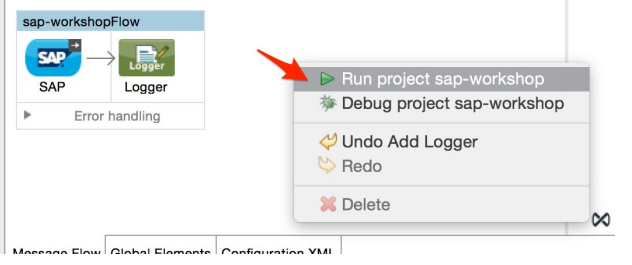
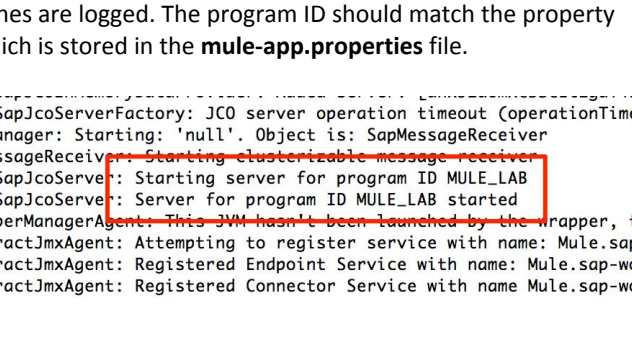
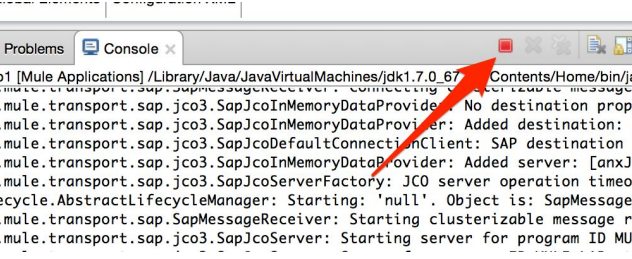
<p>2.1</p>	<p>In the toolkit, search for the SAP Connector by typing in 'SAP' in the Search bar.</p> <p>Drag and drop the SAP Connector onto the canvas.</p> <p>This will set the SAP Connector as an inbound endpoint.</p>	
<p>2.2</p>	<p>If the Select a version... window pops up, select Use newest</p>	
<p>2.3</p>	<p>In the Mule Properties view for the SAP Connector, click on the  icon next to the Connector Configuration field.</p>	
<p>2.4</p>	<p>In the Connection section, enter the following property placeholders into the corresponding fields:</p> <p>AS Host: <code>\${sap.jcoAsHost}</code> User: <code>\${sap.jcoUser}</code> Password: <code>\${sap.jcoPasswd}</code> SAP System Number: <code>\${sap.jcoSysnr}</code> SAP Client: <code>\${sap.jcoClient}</code> Login Language: <code>\${sap.jcoLang}</code></p>	
<p>2.5</p>	<p>In the Required dependencies section, click on Add File and navigate to the src/main/resources folder and select the required dependency file.</p> <p>Be sure to select the correct file for the</p>	

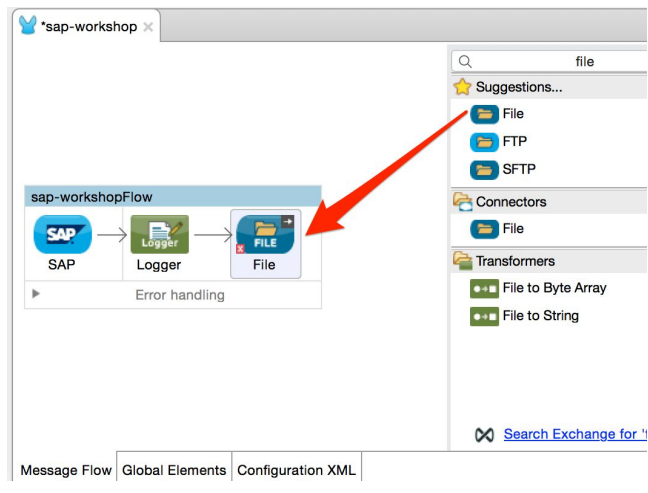
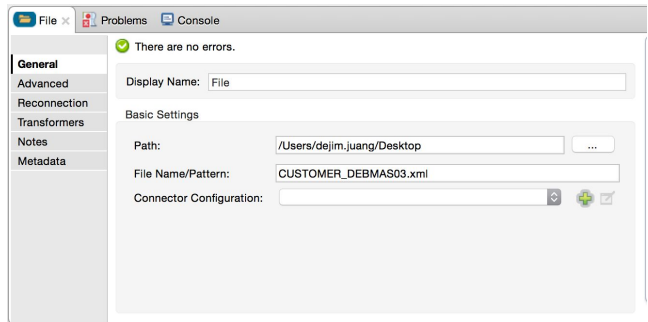
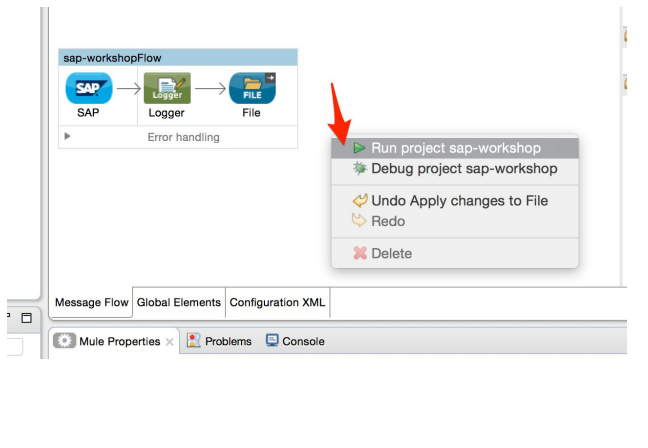
	<p>SAP Java Connector Native Library based on the platform you are using. See below for the list of library by platform.</p> <p>JCo Platform-specific native libraries:</p> <ul style="list-style-type: none"> ● Windows - sapjco3.dll ● Mac OS - libsapjco3.jnilib ● Linux - libsapjco3.so 	
2.6	Next, click on the Advanced tab	
2.7	<p>In the Connection Settings section, enter the following property placeholders into the corresponding fields:</p> <p>Pool Capacity: <code>\${sap.jcoPoolCapacity}</code> Peak Limit: <code>\${sap.jcoPeakLimit}</code></p>	
2.8	Click on Test Connection	
2.9	<p>You should receive the following window if the connection settings are all correct.</p> <p>Click on OK to close the dialog box, and then click OK to close the Global Element Properties window for the SAP Connector</p>	

3. Configure the SAP Connector to receive IDocs

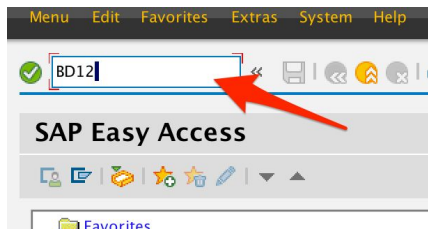
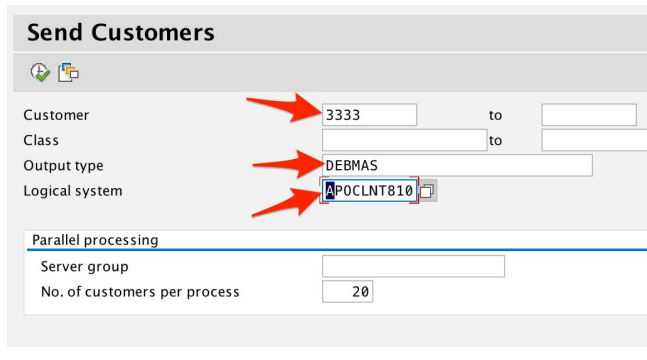

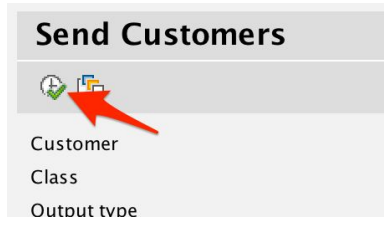

3.1	In the Mule Properties view for the SAP Connector, select IDoc for the Type field and then click on Select...	
-----	---	--

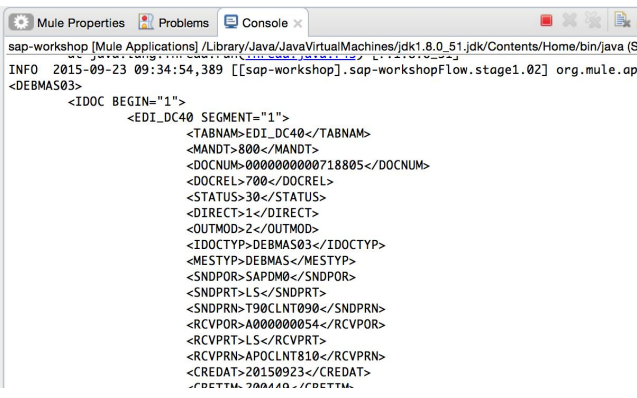

3.2	<div>1. Enter DEBMA03 in the IDoc name field</div> <div>2. Click on Search</div> <div>3. Click on DEBMA03 in the search results.</div> <div>4. Click on Select</div>	 <p>SAP IDoc search</p> <p>SAP IDoc search to select the SAP object. You can also export XML templates/examples or the SAP object XSD.</p> <p>IDoc Filter</p> <p>IDoc name: <input type="text" value="DEBMA03"/> <input type="button" value="Search"/></p> <p>Results (Found 1 results matching 'DEBMA03')</p> <table><thead><tr><th>IDoc</th><th>Description</th></tr></thead><tbody><tr><td>DEBMA03</td><td>Customer master</td></tr></tbody></table> <p><input type="button" value="Cancel"/> <input type="button" value="Select"/></p>	IDoc	Description	DEBMA03	Customer master
IDoc	Description					
DEBMA03	Customer master					
3.3	For the RFC Type field, select Queued RFC (qRFC)	 <p>Object Name: <input type="text" value="DEBMA03 (DEBMA03)"/></p> <p>RFC Type: <input checked="" type="button" value="Queued RFC (qRFC)"/> <input type="button" value="Synchronous RFC (sRFC) (Default)"/> <input type="button" value="Transactional RFC (tRFC)"/></p> <p>XML Version: <input type="button" value="Output XML"/></p>				
3.4	<div>Back in the Mule Properties view, enter the following property placeholders into the JCo Server section.</div> <div>Gateway Host: <code>\${sap.jcoGwHost}</code></div> <div>Gateway Service: <code>\${sap.jcoGwService}</code></div> <div>Program ID: <code>\${sap.jcoProgramId}</code></div> <div>Connection Count: <code>\${sap.jcoConnectionCount}</code></div>	 <p>JCo Server</p> <p>Gateway Host: <input type="text" value="\${sap.jcoGwHost}"/></p> <p>Gateway Service: <input type="text" value="\${sap.jcoGwService}"/></p> <p>Program ID: <input type="text" value="\${sap.jcoProgramId}"/></p> <p>Connection Count: <input type="text" value="\${sap.jcoConnectionCount}"/></p>				
3.5	Under the TID Store section, select Default in Memory TID Store for the Type field.	 <p>Connection Count: <input type="text" value="\${sap.jcoConnectionCount}"/></p> <p>TID Store</p> <p>Transaction ID (TID) stores are used for tRFC and qRFC inbound connections to keep track of current transaction.</p> <p>Type: <input checked="" type="button" value="Default in Memory TID Store"/> <input type="button" value="None (Default)"/> <input type="button" value="Object Store TID Store"/></p> <p>TID Object Store: <input type="text" value="Object Store TID Store"/></p>				
3.6	<div>In the toolkit, search for the Logger Component by typing in 'Log' in the Search bar.</div> <div>Drag and drop the Logger onto the canvas after the SAP Connector.</div>	 <p>sap-workshopFlow</p> <p>SAP → Logger</p> <p>Error handling</p> <p>log</p> <p>Components</p> <p>Logger</p> <p>Search Exchange for 'log'</p>				

3.7	<p>In the Mule Properties view for the Logger, enter the following into the Message field</p> <p><code>#[payload]</code></p>	
3.8	<p>Run the project to test and verify the SAP Connector was configured successfully.</p> <p>Right-click on the canvas and select Run project sap-workshop</p>	
3.9	<p>Review the Console and verify the following lines are logged. The program ID should match the property placeholder value for <code>\$(sap.jcoProgramId)</code> which is stored in the mule-app.properties file.</p> <pre> 169 [main] com.mulesoft.mule.transport.sap.jco3.SapJcoServerFactory: JCO server operation timeout (operationTime 169 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting: 'null'. Object is: SapMessageReceiver 170 [main] com.mulesoft.mule.transport.sap.jco3.SapMessageReceiver: Starting clusterizable message receiver 170 [main] com.mulesoft.mule.transport.sap.jco3.SapJcoServer: Starting server for program ID MULE_LAB 174 [main] com.mulesoft.mule.transport.sap.jco3.SapJcoServer: Server for program ID MULE_LAB started 122 [main] org.mule.module.management.agent.WrapperManagerAgent: This JVM hasn't been launched by the wrapper, t 151 [main] org.mule.module.management.agent.AbstractJmxAgent: Attempting to register service with name: Mule.sap 152 [main] org.mule.module.management.agent.AbstractJmxAgent: Registered Endpoint Service with name: Mule.sap-wo 154 [main] org.mule.module.management.agent.AbstractJmxAgent: Registered Connector Service with name Mule.sap-wo 160 [main] org.mule.DefaultMuleContext: ***** </pre>	
3.10	<p>Click on the Terminate button in the Console tab to stop the project.</p>	

<p>3.11</p>	<p>In the toolkit, search for the File Component by typing in 'File' in the Search bar.</p> <p>Drag and drop the File Component onto the canvas after the Logger.</p>	
<p>3.12</p>	<p>In the Mule Properties view for the File Component, configure the following fields:</p> <p>Path: <Path to Folder></p> <p>File Name/Pattern: CUSTOMER_DEBMAS03.xml</p>	
<p>3.13</p>	<p>The project is configured now. Run the project again and leave it running.</p> <p>Right-click on the canvas and select Run project sap-workshop</p> <p>The flow will act as a JCo Server and listen for IDocs from SAP. When anIDoc with type DEBMAS is sent to your configured Logical System, it will be routed to the Mule flow and generate a file with the data received from SAP.</p> <p>In the next step, we will test this flow.</p>	

4. Test SAP IDoc Inbound Endpoint

4.1	<p>In SAP GUI, enter in SAP transaction code BD12</p>	
4.2	<p>Enter a customer ID in the Customer field. If you don't have one, you can use transaction code XD03</p> <p>In the Output type field, enter DEBMAS</p> <p>In the Logical system field, select the Logical System that was setup as the receiver.</p>	
4.3	<p>Click on Execute  to send the IDoc to Mule.</p>	
4.4	<p>If configured successfully, you should see two Information windows. One is for the setup of the IDoc for the customer, the second is the outbound processing of the IDoc to be sent to the JCo Server.</p>	

4.5	Back in Anypoint Studio , in the Console tab, you should see the Logger output the IDoc for the customer.	 <pre> sap-workshop [Mule Applications] /Library/Java/JavaVirtualMachines/dk1.8.0_51-jdk/Contents/Home/bin/java (S INFO 2015-09-23 09:34:54,389 [[sap-workshop].sap-workshopFlow.stage1.02] org.mule.ap <DEBMA03> <IDOC BEGIN="1"> <EDI_DC40 SEGMENT="1"> <TABNAM>EDI_DC40</TABNAM> <MANDT>800</MANDT> <DOCNUM>000000000718805</DOCNUM> <DOCREL>700</DOCREL> <STATUS>30</STATUS> <DIRECT>1</DIRECT> <OUTMOD>2</OUTMOD> <IDOC TYP>DEBMA03</IDOC TYP> <MESTYP>DEBMA</MESTYP> <SNDPDR>SAPDM0</SNDPDR> <SNDPRT>LS</SNDPRT> <SNDPRN>T90CLNT090</SNDPRN> <RCVPDR>A000000054</RCVPDR> <RCVPRT>LS</RCVPRT> <RCVPRN>APOCLNT810</RCVPRN> <CREDAT>20150923</CREDAT> <CRETIM>200449</CRETIM> <SERIAL>20150923200448</SERIAL> </EDI_DC40> <E1KNA1M SEGMENT="1"> <MSGFN>005</MSGFN> <KUNNR>0000003333</KUNNR> <BBBMR>0000000</BBBMR> </E1KNA1M> </IDOC> </DEBMA03> </pre>
4.6	Lastly, a file should also be created where you configured the Path field in the File Connector .	 <pre> CUSTOMER_DEBMA03.xml 1 <?xml version="1.0"?> 2 <DEBMA03> 3 <IDOC BEGIN="1"> 4 <EDI_DC40 SEGMENT="1"> 5 <TABNAM>EDI_DC40</TABNAM> 6 <MANDT>800</MANDT> 7 <DOCNUM>000000000718805</DOCNUM> 8 <DOCREL>700</DOCREL> 9 <STATUS>30</STATUS> 10 <DIRECT>1</DIRECT> 11 <OUTMOD>2</OUTMOD> 12 <IDOC TYP>DEBMA03</IDOC TYP> 13 <MESTYP>DEBMA</MESTYP> 14 <SNDPDR>SAPDM0</SNDPDR> 15 <SNDPRT>LS</SNDPRT> 16 <SNDPRN>T90CLNT090</SNDPRN> 17 <RCVPDR>A000000054</RCVPDR> 18 <RCVPRT>LS</RCVPRT> 19 <RCVPRN>APOCLNT810</RCVPRN> 20 <CREDAT>20150923</CREDAT> 21 <CRETIM>200449</CRETIM> 22 <SERIAL>20150923200448</SERIAL> 23 </EDI_DC40> 24 <E1KNA1M SEGMENT="1"> 25 <MSGFN>005</MSGFN> 26 <KUNNR>0000003333</KUNNR> 27 <BBBMR>0000000</BBBMR> 28 </E1KNA1M> 29 </IDOC> 30 </DEBMA03> </pre>

Summary

This lab demonstrated how simple it is to setup a JCo Server using Mule and the SAP Connector. As an inbound endpoint, Mule can receive events from SAP and route those objects to other sources easily without extensive coding.

In the next lab, you will learn how to use the SAP Connector as a Function Outbound Endpoint to retrieve a list of customers.

[Lab 2 - Function Outbound Endpoint](#)