

# CONFIGURING, DEPLOYING AND MONITORING WEBUI BASED SCENARIO USING SAP CLOUD PLATFORM INTEGRATION

## Exercise06: SAP Cloud Platform Integration Components Used:

- AS2 Sender Adapter
- SFTP Receiver Adapter
- EDI Splitter
- Router
- EDI to XML Convertor
- XSLT Mapping
- Content Modifier
- Groovy Script
- Mail Receiver Adapter
- AS2 Receiver Adapter

## Exercise06: Prerequisites

- [Install and Configure SAP Cloud Connector](#)

## INTEGRATION SCENARIO

Retail customer has its own EDI Infrastructure (Mendelson Server) and would send purchase order to the partner (Sap Cloud Platform Integration) using ANSI X12 format and AS2 connectivity. Partner has SAP Cloud Platform Integration for EDI Infrastructure and AS2 connectivity. Partner will process EDI document, create a sales order in SAP ERP system using Standard IDOC ORDERS05 and simultaneously send 997 functional acknowledgements to the customer

**Use Case:** In this scenario, there are 2 Parties/Entities involved

- One Retail customer - ElectronicNow(sender) which hosts its own EDI Mandelson platform, where he sends Purchase orders to its partner "SAP" in EDI-X12 format through AS2-Connection to partner's Sap Cloud Platform Integration system and receives the generated functional acknowledgement 997 via AS2.
- Partner's Sap Cloud Platform Integration system(receiver) receives the orders from Retail customer and processes these orders in the SAP CPI Backend and generate IDOC ORDERS05 sales orders.

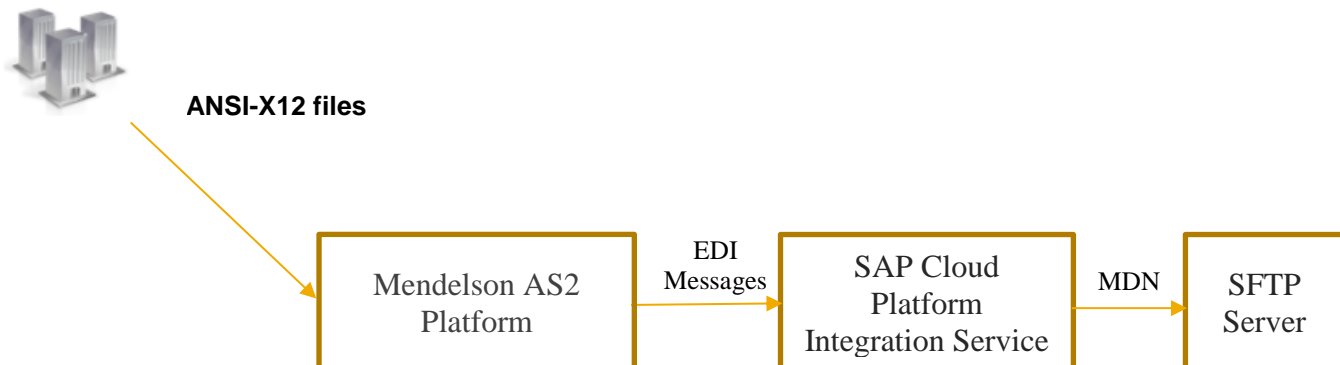
Also, ElectronicNow Sender compresses, signs (Senders' private key) and encrypts (Receivers' public key) while sending the EDI message. At Receiver's side, the receiver decrypts (Receivers' private key), verifies the signature (Senders' public key) and finally decompresses the EDI message.

Here we are using private key **Key2** of AS2 Mendelson for signing and a public certificate from SAP Cloud Integration tenant for encryption purpose when we send EDI messages to partner

For the AS2 secure message exchange to happen between Mendelson and SAP Cloud Platform Integration tenant, we need to exchange the public keys between these two applications.

**Exercise 06a** ElectronicNow Customer will send ANSI X12 850 file from Mendelson AS2 Simulator to its partner SAP Cloud Platform Integration tenant as EDI messages using AS2 connection and SAP CPI system will process the EDI messages and generate MDN attachments

Customer ElectronicNow

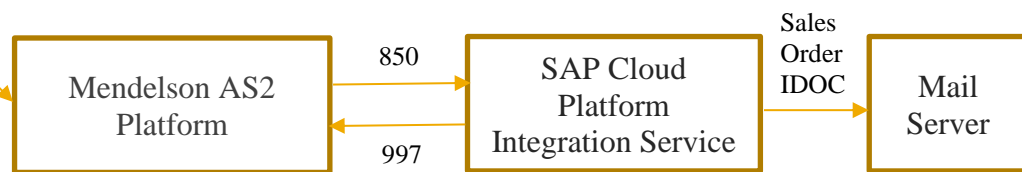


**Exercise 06b** ElectronicNow Customer will send ANSI X12 850 file to Mendelson AS2 platform, which will send it to SAP Backend CPI System as 850 EDI messages with Compression, Signing, and Encryption, SAP CPI service will process it and creates IDOC Sales orders and send the customer with functional acknowledgement of 997 message. Here we showcase both Synchronous MDN and Asynchronous MDN with Sign

Customer ElectronicNow



EDI-X12 files

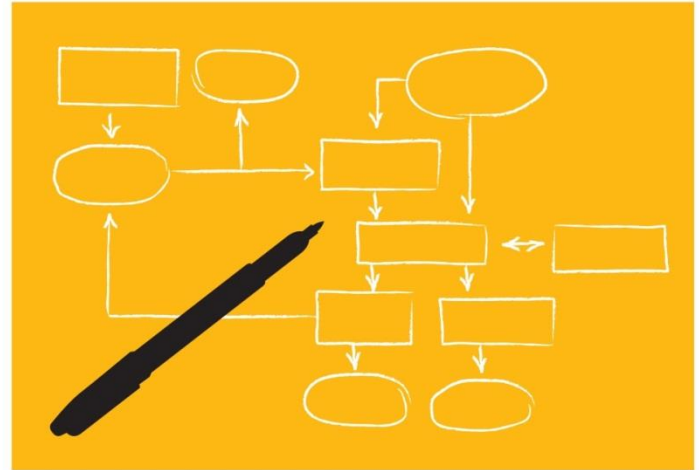


Welcome to the Integration Flow challenge!

In this exercise, your aim is to create an integration flow that solves a challenge (*described in the Integration scenario*).

And when you work your way through the exercise, our aim is that you learn:

1. How to access and work with the Cloud Platform Integration Web tooling
2. How to speed up your integration project by leveraging reference integration flows
3. How to customize a reference integration flow by configuring its connectivity and flow steps
4. Basic monitoring of an integration flow
5. Using Open-source tools to test your integration flow



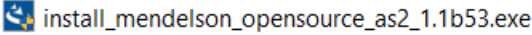
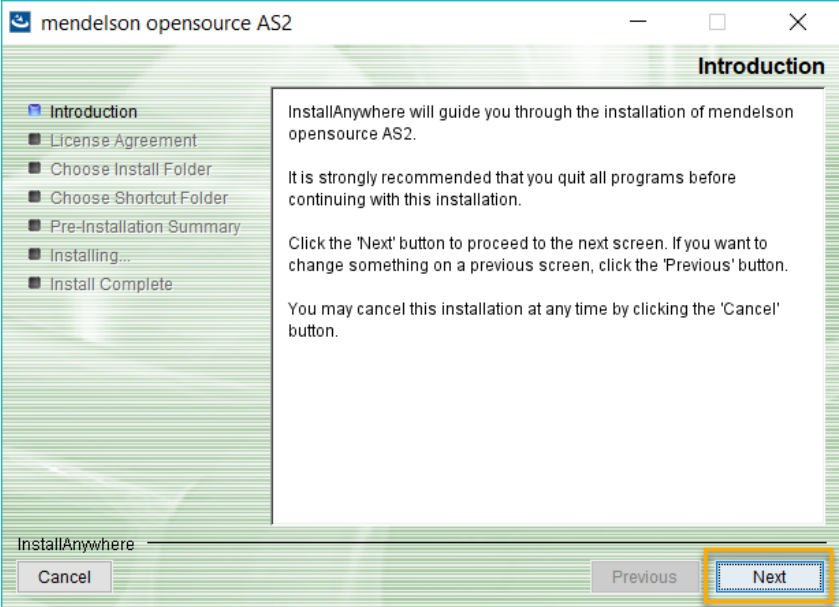
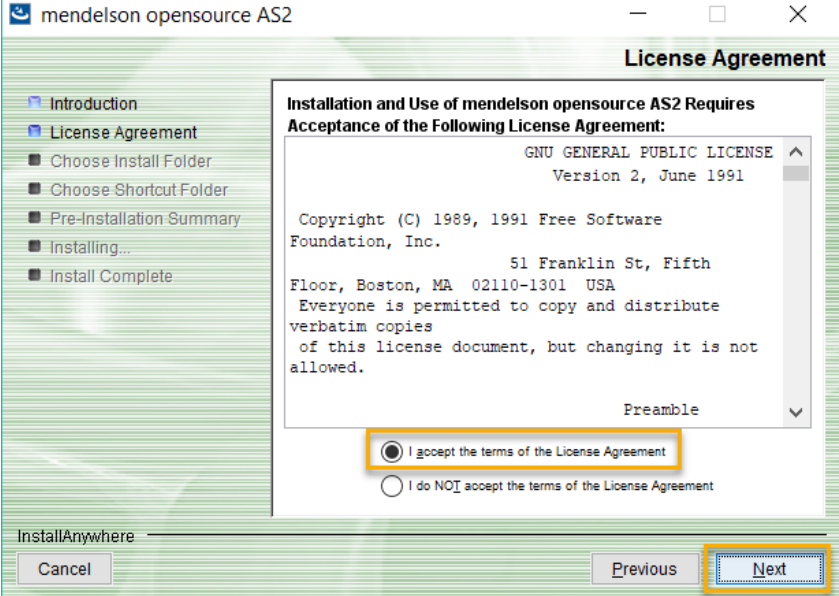
## Exercise Files

Exercise Files are provided by the session moderator. Download and Extract them into one of your local folders for use later in your exercises and setup:

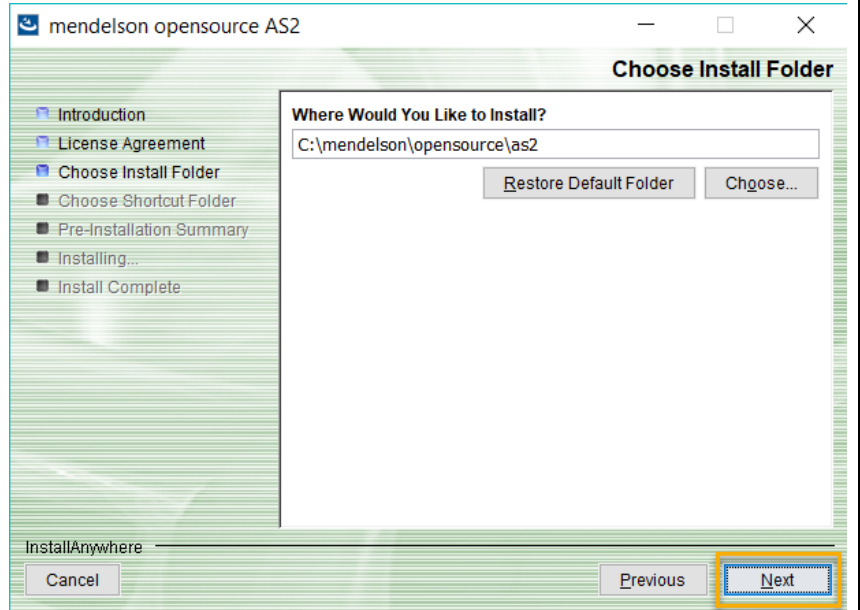
### Note

1. In the exercise, we have used the notation XX to refer to the content created by you or your group uniquely. The session moderator shall assist you with assigning you/group the unique ID that you can use to replace XX.
2. In the exercise, we have used **rktw002/rktw003** as the example tenants. The final tenant details shall be provided by the session moderator.
3. Please note that colours and other visual appearance might differ slightly from the screenshot screens, as the CPI editor might have received feature upgrades since production of this content.

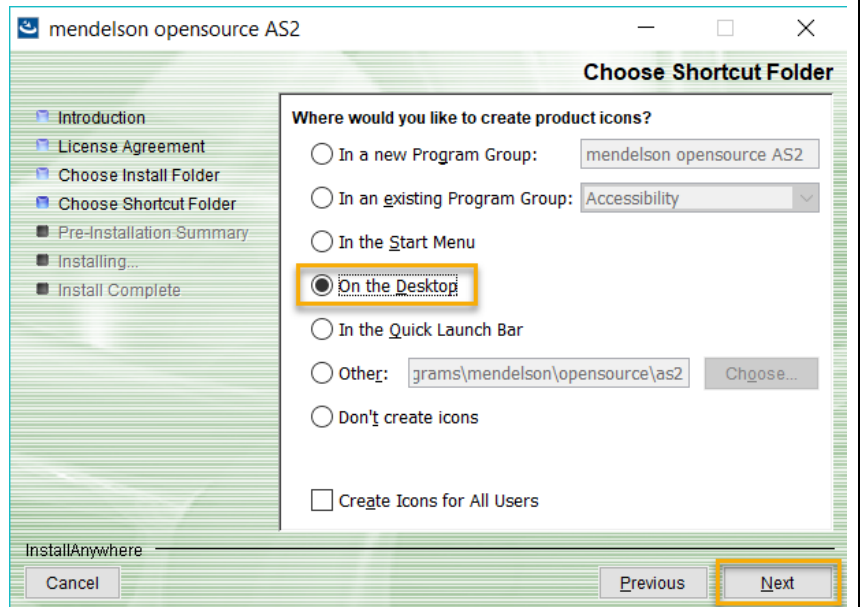
## EXERCISE 06A PREREQUISITE (MENDELSON AS2 SETUP)

Explanation	Screenshot
Follow these steps to install Mendelson AS2 tool and configure customer and partners	
1) Run the <b>install_mendelson_opensource_as2_1.1b53.exe</b> from file system share or from <a href="http://mendelson-e-c.com/as2/">http://mendelson-e-c.com/as2/</a>	
2) Click on <b>Next</b>	
3) Accept the license agreement and click on <b>Next</b>	

- 4) Select the install location and click **Next**

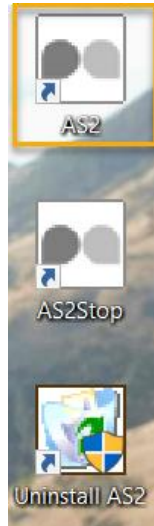


- 5) Select **“on the Desktop”** option for create product icons and click on **Next** and complete the installation



- 6) After Installation, you should see **AS2** and **AS2Stop** shortcuts created on the desktop.

Launch **AS2** server by clicking on **Run as administrator**

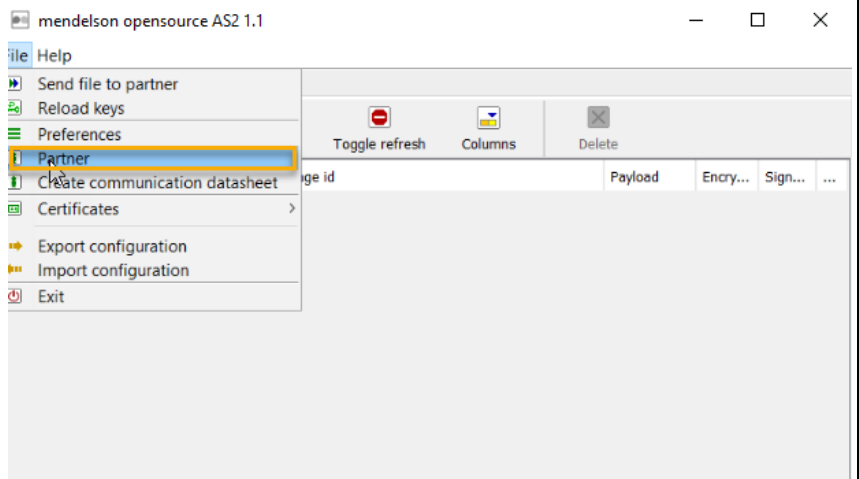


- 7) Enter <http://localhost:8080/as2/HttpReceiver> in your browser and you should see the below message.



- 8) Let's configure the AS2 partner i.e. ElectronicNow who act as initiator to post AS2 messages to its partner SAP.

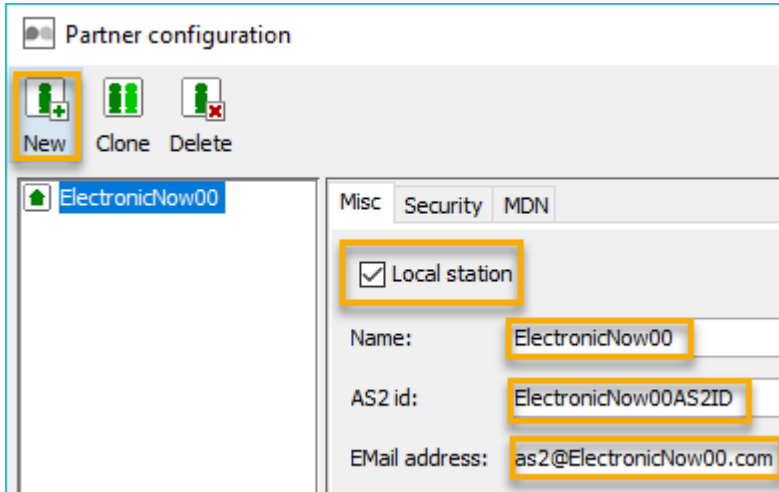
Select **File -> Partner**



9) Click on **New** and select **Local Station** and configure the following values:

- a. **Name:** ElectronicNowXX
- b. **AS2 id:** ElectronicNowXXAS2ID
- c. **Email address:**  
[as2@ElectronicNowXX.com](mailto:as2@ElectronicNowXX.com)

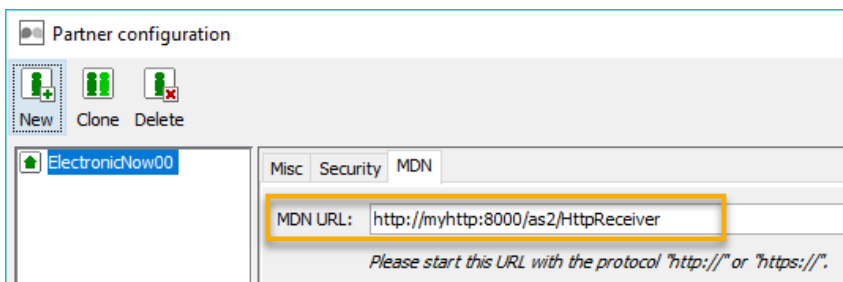
**Note:** Replace **XX** with your participant id



The screenshot shows the 'Partner configuration' dialog box. The 'New' button is highlighted. Below it, 'ElectronicNow00' is selected. The 'Misc' tab is active, and the 'Local station' checkbox is checked. The fields for Name, AS2 id, and Email address are filled with 'ElectronicNow00', 'ElectronicNow00AS2ID', and 'as2@ElectronicNow00.com' respectively.

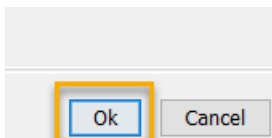
10) Choose the **MDN** tab and enter the MDN URL as  
`http://<Virtual Host>:<Virtual Port>/as2/HttpReceiver`

**Note:** Here **Virtual Host** and **Virtual Port** are the virtual host and port addresses which you have entered in Cloud Connector.



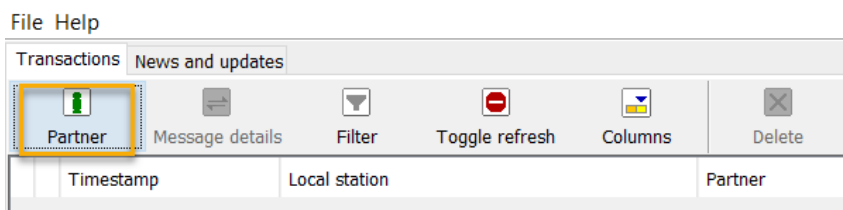
The screenshot shows the 'Partner configuration' dialog box with the 'MDN' tab selected. The 'MDN URL' field is highlighted and contains the text 'http://myhttp:8000/as2/HttpReceiver'. A note below the field states: 'Please start this URL with the protocol "http://" or "https://".'

11) Click **Ok** to save the configuration



The screenshot shows the 'Ok' and 'Cancel' buttons from the dialog box, with the 'Ok' button highlighted.

12) Now let's configure ElectronicNow's AS2 partner i.e. SAPCompany. Select **Partner** option again



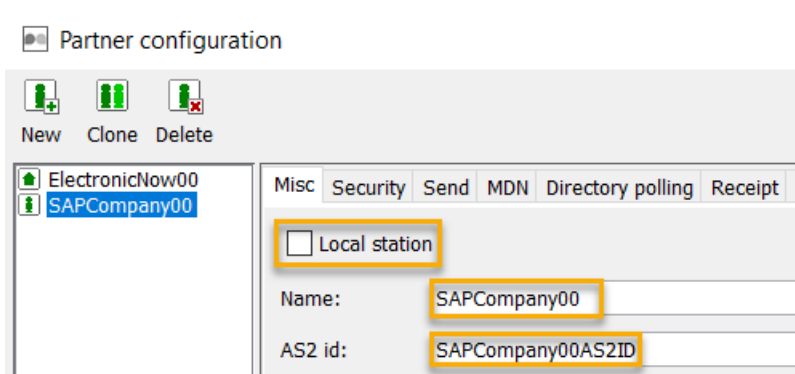
The screenshot shows the 'File' menu with the 'Partner' option highlighted. Below the menu, there are buttons for 'Message details', 'Filter', 'Toggle refresh', 'Columns', and 'Delete'. A table below these buttons has columns for 'Timestamp', 'Local station', and 'Partner'.

13) Click on **New** and configure the following values:

- a. **Name:** SAPCompanyXX
- b. **AS2 id:** SAPCompanyXXAS2ID

Make sure **Local Station** is unchecked

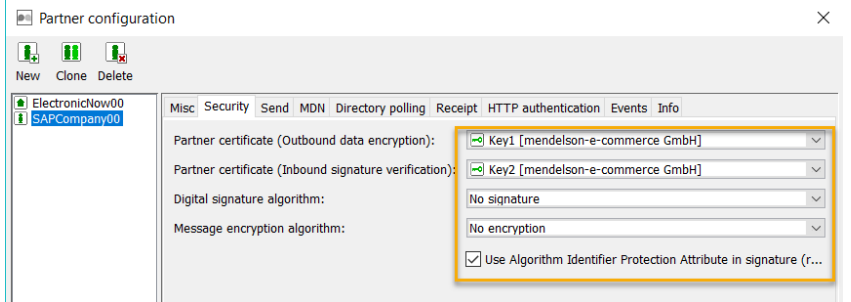
**Note:** Replace **XX** with your participant id



The screenshot shows the 'Partner configuration' dialog box. The 'New' button is highlighted. Below it, 'SAPCompany00' is selected. The 'Misc' tab is active, and the 'Local station' checkbox is unchecked. The fields for Name and AS2 id are filled with 'SAPCompany00' and 'SAPCompany00AS2ID' respectively.



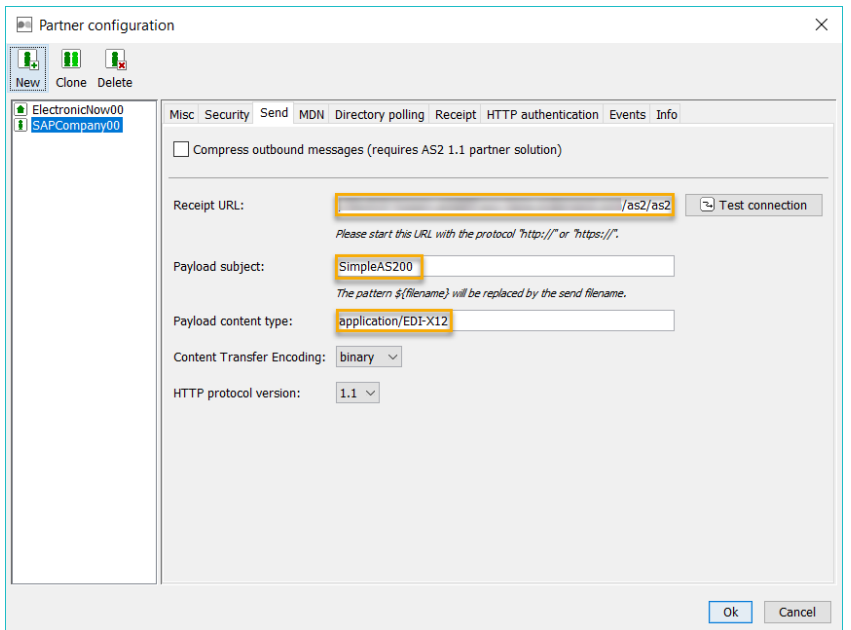
- 14) Choose the **Security** tab and configure security settings as shown in the screenshot.



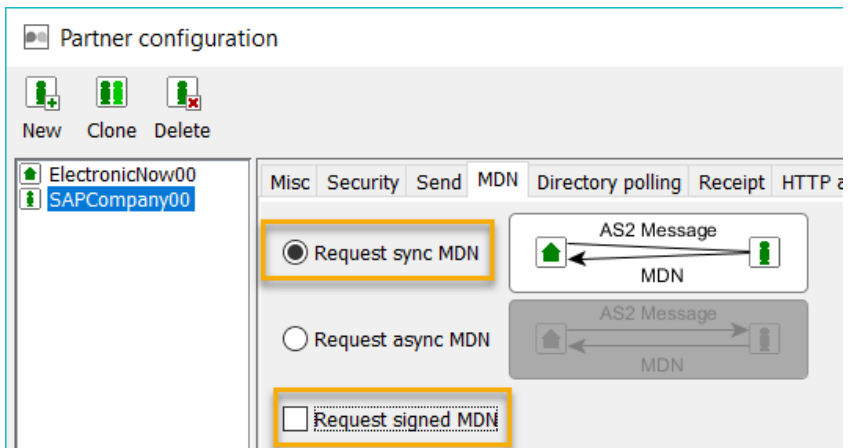
- 15) Choose the **Send** tab and configure settings as given below:

- Receipt URL:** <Runtime URL of the SAP Cloud Integration tenant>/as2/as2
- Payload subject:** SimpleAS2XX
- Payload content type:** application/EDI-X12

**Note:** Replace **XX** with your participant id



- 16) Choose the **MDN** tab and Configure MDN settings as shown in the screenshot. Uncheck "Request signed MDN"



17) Choose the **HTTP authentication** tab and configure HTTP authentication to send AS2 messages to Cloud Platform Integration tenant as shown in the screenshot.

Where Username and Password is your Cloud Platform Integration tenant login user credentials.

Partner configuration

New Clone Delete

ElectronicNow00  
SAPCompany00

Misc Security Send MDN Directory polling Receipt HTTP authentication

☒ Use HTTP authentication to send AS2 messages

Username:

Password:

☐ Use HTTP authentication to send async MDN

Username:

Password:

18) Click **Ok** to save the configuration

Ok Cancel

19) Configure proxy settings (Optional)

Navigate to **File -> Preferences -> Proxy** and set the proxy value.

If you are in internet, you may leave these settings blank. If your organizations require it, then you can mention proxy settings as per your organization.

Preferences

Misc Proxy Security Directories Maintenance Notification Language

☒ Use a proxy for outgoing HTTP/HTTPS connections

Proxy URL:  :

☐ Use proxy authentication

Proxy login user:

Proxy login pass:

Ok

20) Import Cloud Platform Integration root certificate on to the Mandelson tool

a. In the Mandelson tool navigate to **File -> Certificates -> SSL/TLS**

b. Click on import certificate

mendelson opensource AS2 1.1

File Help

Send file to partner  
Reload keys  
Preferences  
Partner  
Create communication datasheet  
Certificates  
Display HTTP server configuration  
Exit

Sign/Crypt  
SSL/TLS

Toggle refresh Columns Delete

Partner

Available certificates and keys (SSL) | JKS

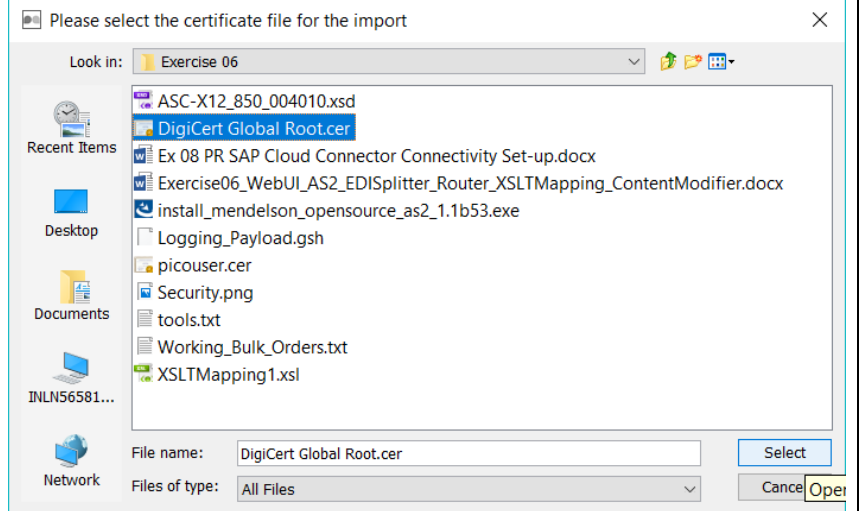
File Import Export Tools

Import certificate Rename alias Delete key/certificate

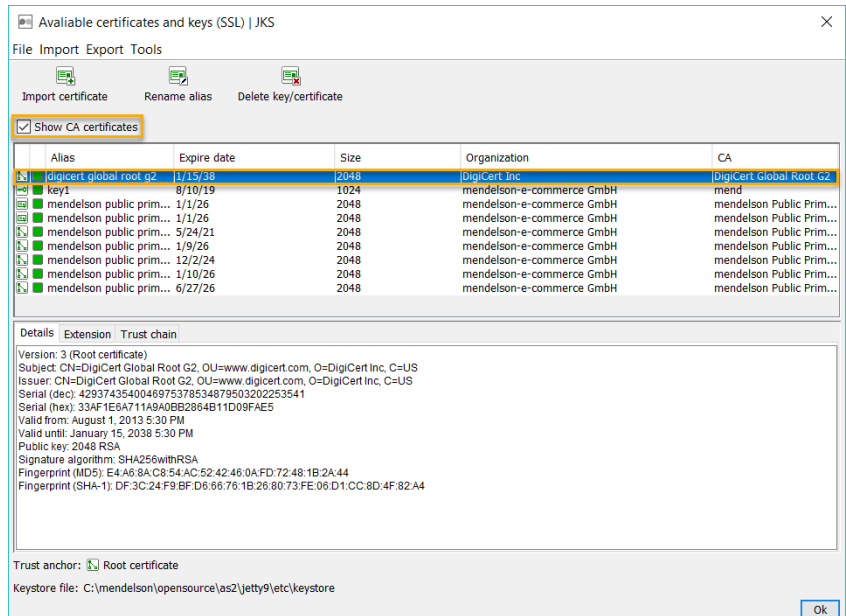
☐ Show CA certificates

Alias	Expire date	Size	Organization	CA
key1	8/10/19	1024	mandelson-e-commerce GmbH	mand

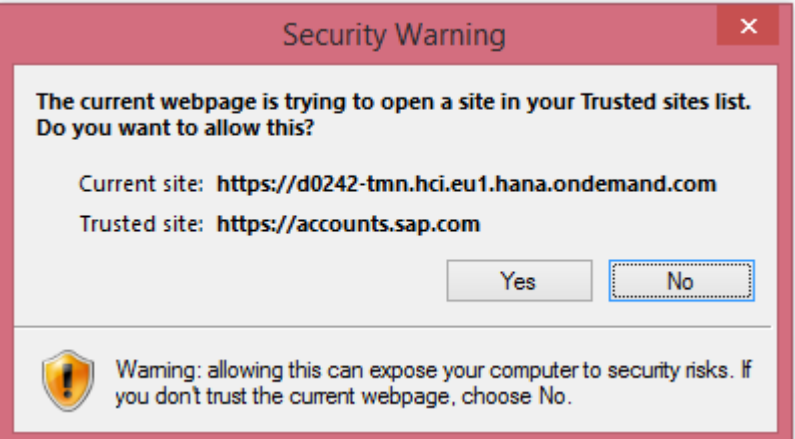
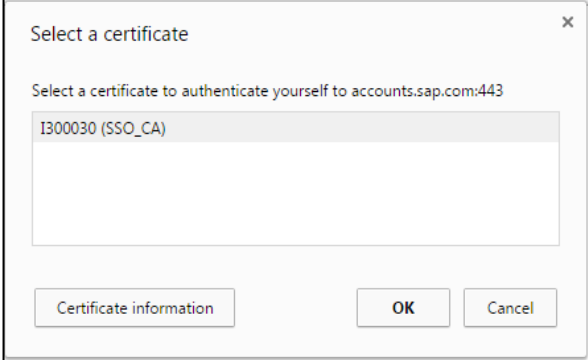
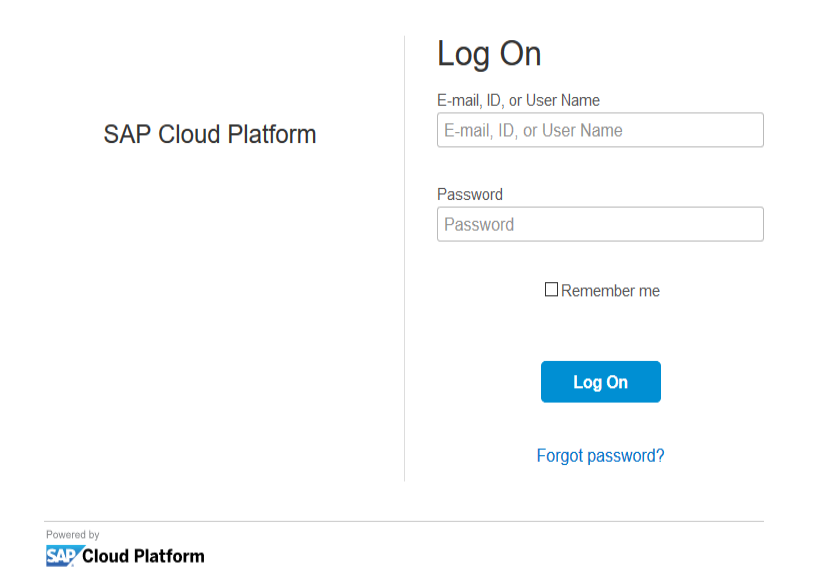
- c. Click on Browse and from the Exercise folder, import the certificate **DigiCert Global Root.cer** file.



- 21) Check **"Show CA certificates"** to view Root certificate. SAP Cloud Platform Integration Root certificate should be listed.



## EXERCISE 06A

Explanation	Screenshot
<p><b>Follow these steps for integration flow creation in Web UI.</b></p>	
<p>1) Access the Web application using Google Chrome (or IE10 Browser)</p> <ol style="list-style-type: none"> <li>Open Internet Explorer</li> <li>Copy the application URL or the URL that you can use will be assigned to you by the instructor.</li> <li>You will get a security warning if you want to open this site in your trusted sites list.</li> </ol> <p>Accept it with 'Yes'</p>	 <p>A security warning dialog box titled "Security Warning" with a red close button. The message reads: "The current webpage is trying to open a site in your Trusted sites list. Do you want to allow this?". It shows the "Current site: https://d0242-tmn.hci.eu1.hana.ondemand.com" and the "Trusted site: https://accounts.sap.com". There are "Yes" and "No" buttons. Below the buttons is a warning icon and text: "Warning: allowing this can expose your computer to security risks. If you don't trust the current webpage, choose No."</p>
<p>2) If you get displayed your Certificate List, click on 'Cancel'</p>	 <p>A "Select a certificate" dialog box with a close button. It says "Select a certificate to authenticate yourself to accounts.sap.com:443". A list box contains "I300030 (SSO_CA)". At the bottom are buttons for "Certificate information", "OK", and "Cancel".</p>
<p>3) You will now see the log-on screen for the SAP ID Service/SAP Cloud Platform.</p> <p>Enter the following credentials:</p> <p><i>User:</i> &lt;Provided by instructor&gt;  <i>Password:</i> &lt;Provided by instructor&gt;</p>	 <p>The SAP Cloud Platform log-on screen. It features the "SAP Cloud Platform" logo on the left. On the right, under the heading "Log On", there are input fields for "E-mail, ID, or User Name" and "Password". Below these is a "Remember me" checkbox and a blue "Log On" button. A link for "Forgot password?" is at the bottom right. At the very bottom, it says "Powered by SAP Cloud Platform" with the SAP logo.</p>

- 4) Now you will have access to the Web application. Familiarize yourself with the environment and choose the **Design** area. Here, you will create an Integration Package to store your iFlows for different Exercises

Click on **Create**

Enter Details

- a) **Name:** Cloud Platform, Integration Service Workshop (Exercises) Group\_XX

**NOTE:** Replace XX with the group number provided by the instructor

- b) **Short Description:** EKT 2018 Cloud Platform, Integration Service
- c) **Version:** 1.0.0
- d) **Vendor:** SAP

The screenshot shows the SAP Cloud Platform Integration Design area. The left sidebar has a menu with 'Discover', 'Design', 'Monitor', and 'Settings'. The 'Design' tab is selected. The main area shows the 'Design / New Package' form. The form has a header with 'Design / New Package /' and a 'Create' button. The form title is 'Cloud Platform, Integration Service Workshop (Exercises) Group\_00'. Below the title are tabs for 'Header', 'Overview', 'Artifacts', 'Documents', and 'Tags'. The 'Header' tab is active. The form fields are: '\*Name:' (Cloud Platform, Integration Service Workshop (Exercises) Group\_00), '\*Technical Name:' (CloudPlatformIntegrationServiceWorkshopExercisesGroup00), '\*Short Description:' (EKT Cloud Platform, Integration Service), 'Version:' (1.0.0), and 'Vendor:' (SAP).

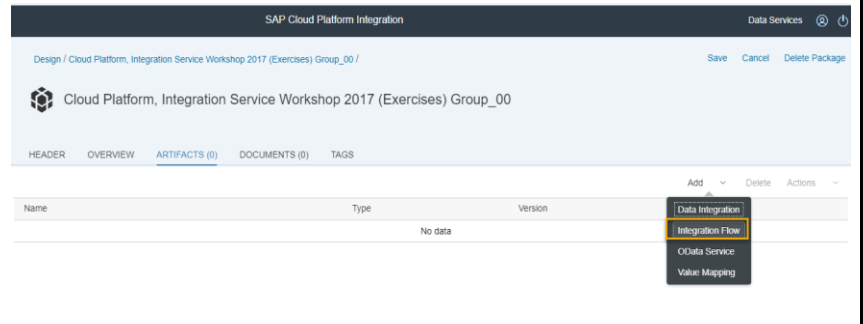
- 5) Click on Save

The screenshot shows the SAP Cloud Platform Integration Design area. The left sidebar has a menu with 'Discover', 'Design', 'Monitor', and 'Settings'. The 'Design' tab is selected. The main area shows the 'Design / Cloud Platform, Integration Service Workshop (Exercises) Group\_00 /' form. The form title is 'Cloud Platform, Integration Service Workshop (Exercises) Group\_00'. Below the title are tabs for 'Header', 'Overview', 'Artifacts', 'Documents', and 'Tags'. The 'Header' tab is active. The form fields are: '\*Name:' (Cloud Platform, Integration Service Workshop (Exercises) Group\_00), '\*Technical Name:' (CloudPlatformIntegrationServiceWorkshopExercisesGroup00), '\*Short Description:' (EKT Cloud Platform, Integration Service), 'Version:' (1.0.0), and 'Vendor:' (SAP). The 'Save' button is highlighted.

- 6) Navigate to **Artifacts** Tab

The screenshot shows the SAP Cloud Platform Integration Design area. The left sidebar has a menu with 'Discover', 'Design', 'Monitor', and 'Settings'. The 'Design' tab is selected. The main area shows the 'Design / Cloud Platform, Integration Service Workshop (Exercises) Group\_00 /' form. The form title is 'Cloud Platform, Integration Service Workshop (Exercises) Group\_00'. Below the title are tabs for 'Header', 'Overview', 'Artifacts', 'Documents', and 'Tags'. The 'Artifacts' tab is active.

7) Click on **Add-> Integration Flow**



- 8) Select **Create** and enter following details:
- a. Name: **Exercise06a\_XX**
  - b. Description: **Exercise06a\_WebUI\_AS2\_SFTP**
  - c. Sender: **AS2**
  - d. Receiver: **SFTP**
  - e. Click on **OK**

Add integration flow artifact

☒ Create ☐ Upload

\*Name:

\*ID:

Product Profile:

Description:

Sender:  Receiver:

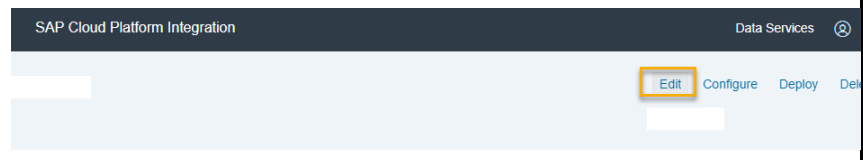
9) Click on **Exercise06a\_XX**



### Define and edit integration flow

### Follow steps to edit the integration flow

10) Click on **Edit** button on the upper corner on right hand side.



11) Click on Sender system and set Name as **AS2**



The screenshot shows a configuration window for a system named 'AS2'. On the left is a vertical toolbar with icons for navigation and actions. The main area contains a box labeled 'AS2' with a dashed orange border. To the right of this box are three circular icons: an information icon, a right-pointing arrow, and a trash icon. Below the box, the word 'System' is displayed. At the bottom, there is a label '\*Name:' followed by a text input field containing the value 'AS2'.

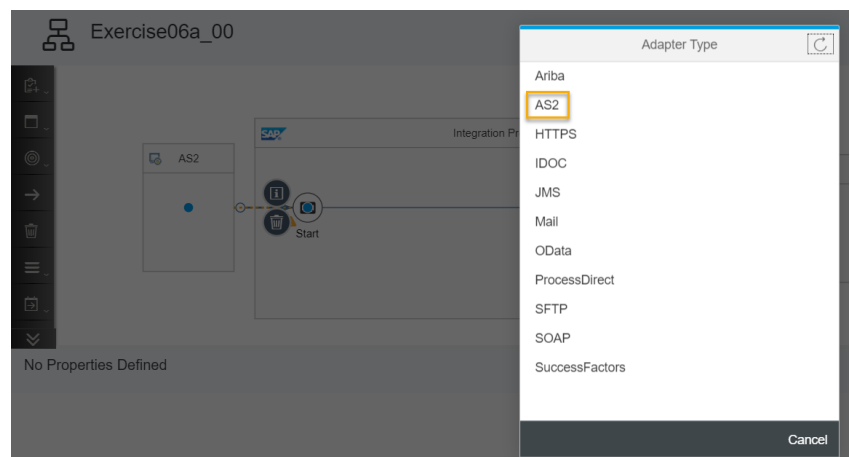
12) Click on Receiver System and set Name as **SFTP**



The screenshot shows a configuration window for a system named 'SFTP'. It features a similar layout to the previous one, with a vertical toolbar on the left, a box labeled 'SFTP' with a dashed orange border, and three circular icons (information, arrow, trash) to its right. Below the box is the label 'System'. At the bottom, the label '\*Name:' is followed by a text input field containing the value 'SFTP'.

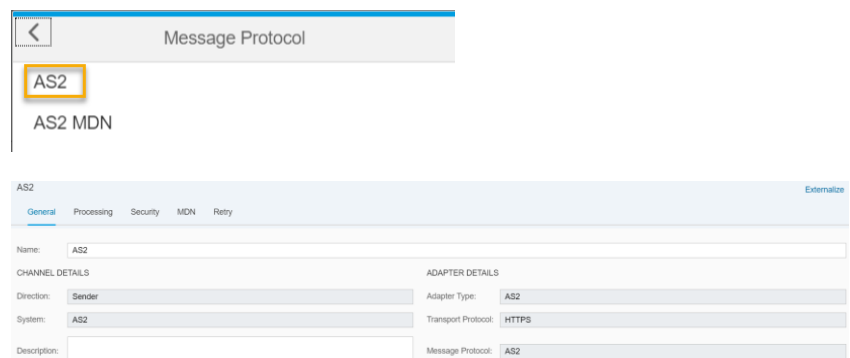
13) Click on **AS2** sender system, select **Connector** and drag it to **Start** message step.  
This would open the sender adapter list.

Select **AS2** adapter



The screenshot displays an integration process editor for 'Exercise06a\_00'. It shows a flow diagram with a box labeled 'AS2' and a 'Start' message step. A context menu is open over the 'AS2' box, listing various adapter types: Ariba, AS2, HTTPS, IDOC, JMS, Mail, OData, ProcessDirect, SFTP, SOAP, and SuccessFactors. The 'AS2' option is highlighted with a dashed orange border. A 'Cancel' button is visible at the bottom right of the menu.

Select **AS2** as Message Protocol



The screenshot shows the 'Message Protocol' configuration window for 'AS2 MDN'. It has a title bar with a back arrow and the text 'Message Protocol'. Below the title bar, 'AS2' is selected in a list, and 'AS2 MDN' is displayed below it. The window is divided into two main sections: 'CHANNEL DETAILS' and 'ADAPTER DETAILS'. The 'CHANNEL DETAILS' section includes fields for 'Name' (AS2), 'Direction' (Sender), 'System' (AS2), and 'Description'. The 'ADAPTER DETAILS' section includes fields for 'Adapter Type' (AS2), 'Transport Protocol' (HTTPS), and 'Message Protocol' (AS2). There are tabs for 'General', 'Processing', 'Security', 'MDN', and 'Retry', with 'General' being the active tab. An 'Externalize' button is located in the top right corner.

14) Switch to **Processing** tab, enter the following details:

- a. Message ID Left Part: \*
- b. Message ID Right Part: \*
- c. Partner AS2 ID:  
**ElectronicNowXXAS2ID**
- d. Own AS2 ID:  
**SAPCompanyXXAS2ID**
- e. Message Subject:  
**SimpleAS2XX**

Keep other properties related to Security, MDN and Retry to the default values

**Note:** Replace **XX** with your participant id

Save

## AS2

General

**Processing**

Security

MDN

Retry

### EXPECTED MESSAGES

\*Message ID Left Part:

\*

\*Message ID Right Part:

\*

\*Partner AS2 ID:

ElectronicNow00AS2ID

\*Own AS2 ID:

SAPCompany00AS2ID

\*Message Subject:

SimpleAS200

\*Number of Concurrent Processes:

1

\*Authorization:

User Role

\*User Role:

ESBMessaging.send

### MESSAGE SETTINGS

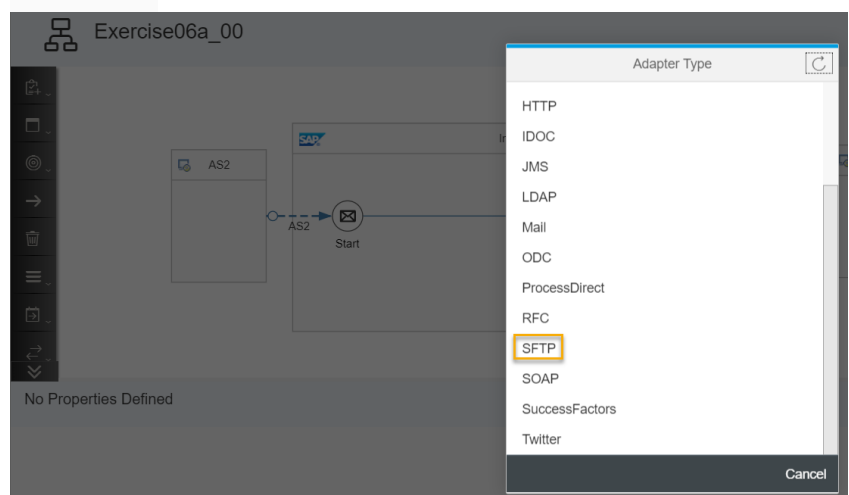
☐ Mandatory File Name

☐ Duplicate Message ID

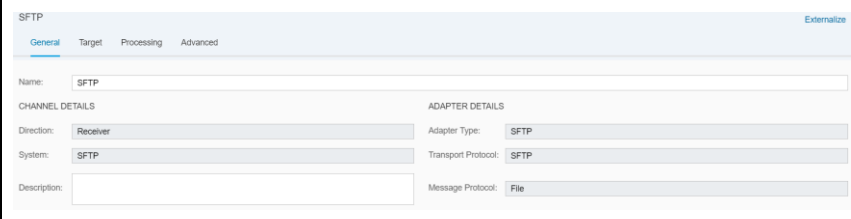
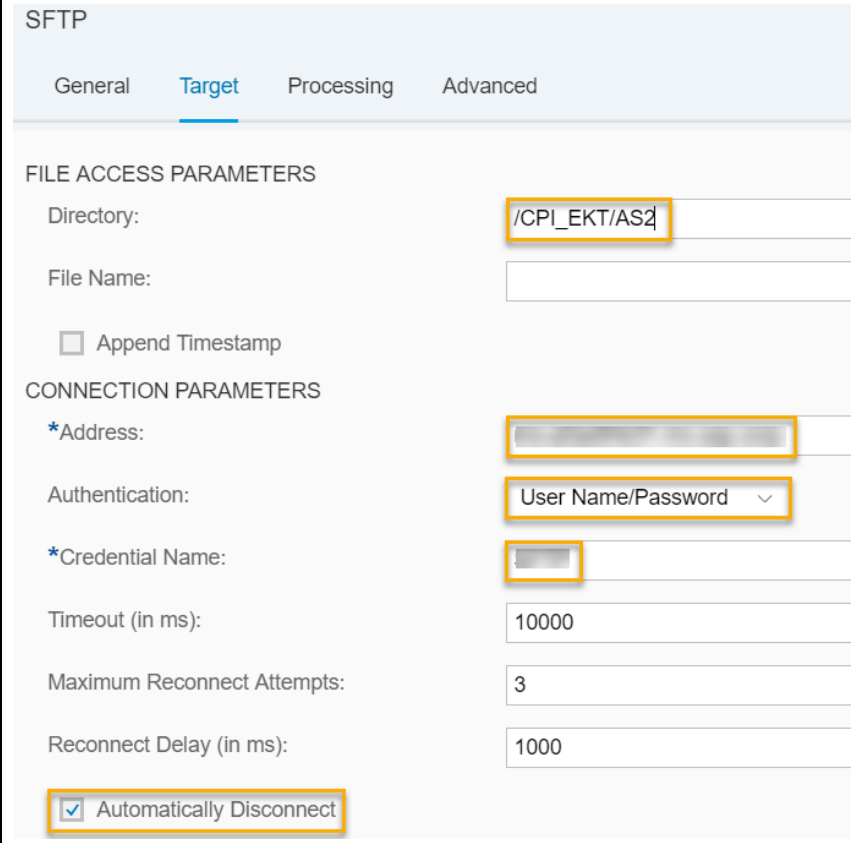
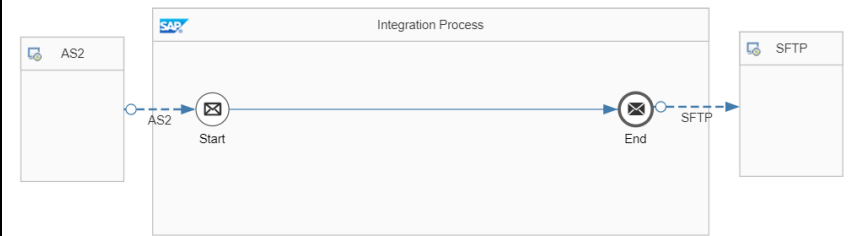
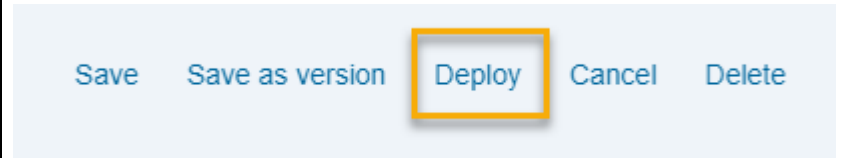
☐ Duplicate File Name

15) Click on **End** message step, select **Connector** and drag it to **SFTP** receiver system. This would open the receiver adapter list.

Select **SFTP** adapter





	
<p>16) Switch to <b>Target</b> tab, enter the following details:</p> <ol style="list-style-type: none"> <li><b>Directory:</b> /CPI_EKT/AS2</li> <li><b>Address:</b> &lt;provided by the Instructor&gt;</li> <li><b>Authentication:</b> User Name/Password</li> <li><b>Credential Name:</b> Name of credential object which holds SFTP user and password, deployed on the tenant.</li> <li><b>Automatically Disconnect:</b> checked</li> </ol> <p>Save</p>	
<p>17) Your integration flow should look like this.</p>	
<b>Deploy Integration Project on tenant:</b>	<b>Follow this step to deploy Integration Project on tenant.</b>
<p>18) Press <b>Deploy</b> in the Integration Flow Task Bar.</p>	

- 19) You will receive a confirmation.  
Click on **Yes**.

Once the deployment is completed successfully you will receive a 2<sup>nd</sup> notification.

?

Confirmation

Do you want to deploy?

YesNo

i

Deployment

'Exercise06a\_00' is triggered for deployment.

OK

- 20) Verify if the deployment is successful:

In the first level Menu Bar switch to Section **Monitor** and then click on **All** in **Manage Integration Content**.

You should see an entry with your integration flow.

Check the 'status'. It should be in status **Started**.

Discover

Design

Monitor

Se Operations view

Manage Integration Content

AllAll

1All1Started

Exercise06a\_00

Integration Flow

Started

- 21) A Message queue named after the integration flow is created under **Monitoring -> Manage Stores -> Message Queues**

Manage Stores

Data StoresVariablesMessage Queues

1Stores0Variables1Queues

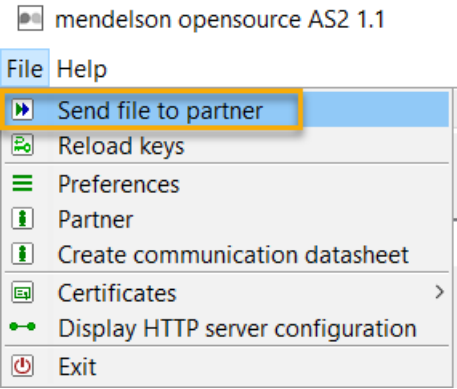
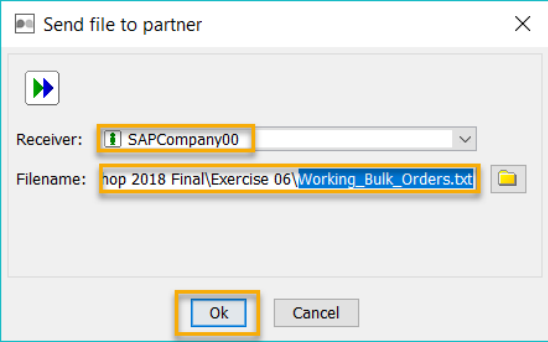
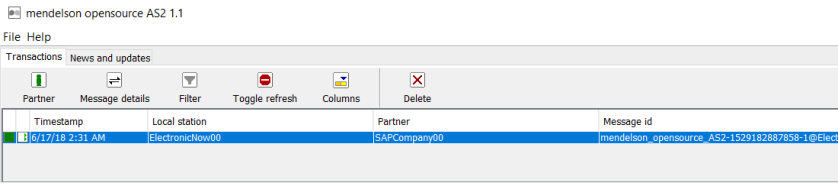
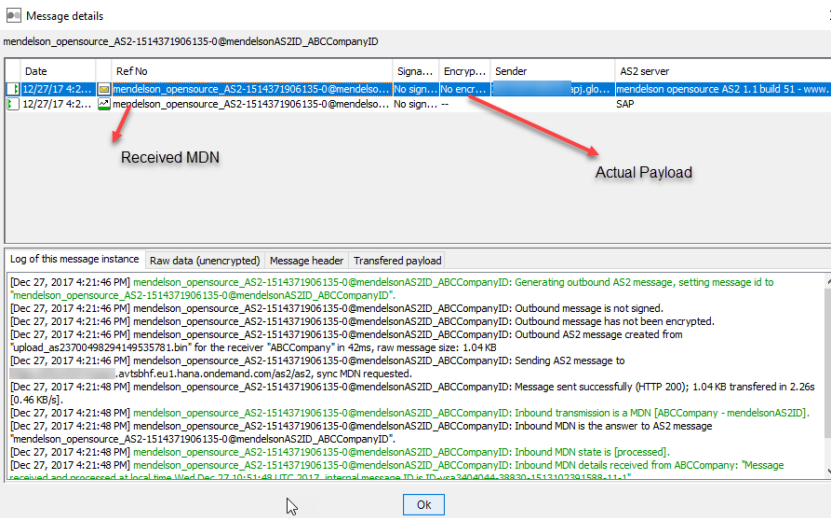
Overview / Manage Message Queues

i

Used Capacity: 0.0 / 9437184 KB

Queues (1)Filter by NameSearchActions

Name	Entries	Size (in MB)
AS2_Exercise06a_00_AS2_74149237_f0f3_3d62_ae62_b8a6d8718241	0	0.000

Execute end to end scenario:	Follow steps to execute end to end scenario.
<p>22) Let's test a simple AS2 message post from Mendelson to SAP Cloud Platform Integration Tenant</p> <p>In the Mendelson tool, Choose <b>File -&gt; Send file to partner</b></p>	
<p>23) Choose <b>SAPCompanyXX</b> from the Receiver dropdown and select <b>Working_Bulk_Orders.txt</b> file and click Ok</p>	
<p>24) You should see a successful message in the Mandelson tool as shown below</p>	
<p>25) Double click on the message to see the message that was sent and the MDN received back</p>	

26) On the CPI tenant, Navigate to **Monitor Message processing** to see a Completed message with an MPL attachments (**MDN Attachment**)

The screenshot shows the SAP CPI Monitor Message processing interface. On the left, a table lists messages. The first message, 'Exercise06a\_00', is highlighted and has a 'Completed' status. The second message, also 'Exercise06a\_00', is also 'Completed'. On the right, the details for the selected message are shown. The 'Attachments' tab is active, displaying a table with one attachment: 'MDN Attachment' of type 'text/xml'. Below this, the 'Log' tab is active, showing the message log. The log contains the following text:

```

-----=_Part_11_1304829430.1529183689987
Content-type: text/plain
Content-Transfer-Encoding: 7bit

Message received and processed at local time Sat Jun 16 21:14:49 UTC
-----=_Part_11_1304829430.1529183689987
Content-Type: message/disposition-notification
Content-Transfer-Encoding: 7bit

Reporting-UA: SAP AS2 Adapter
Original-Recipient: rfc822; SAPCompany00AS2ID
Final-Recipient: rfc822; SAPCompany00AS2ID
Original-Message-ID: <mendelson_opensource_AS2-1529183687137-2@Elect
Disposition: automatic-action/MDN-sent-automatically; processed

-----=_Part_11_1304829430.1529183689987--

```

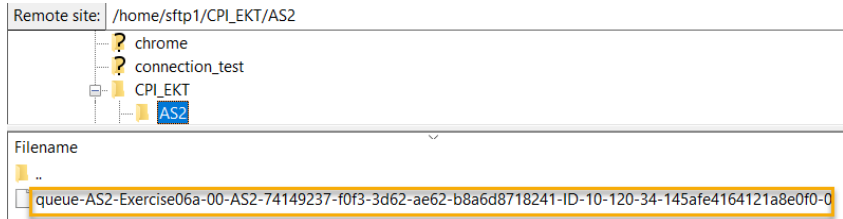
**Check generated file with FileZilla**

**Follow these steps to check the file in the SFTP folders**

27) Open FileZilla and connect to the **SFTP Server** using the credentials provided by your instructor

The screenshot shows the FileZilla Site Manager dialog box. The 'General' tab is selected. The 'Host' field is set to '<Provided by your instructor>'. The 'Protocol' is set to 'SFTP - SSH File Transfer Protocol'. The 'Logon Type' is set to 'Normal'. The 'User' and 'Password' fields are both set to '<Provided by your instructor>'. The 'Background color' is set to 'None'. The 'Comments' field is empty. At the bottom, the 'Connect' button is highlighted with a red box.

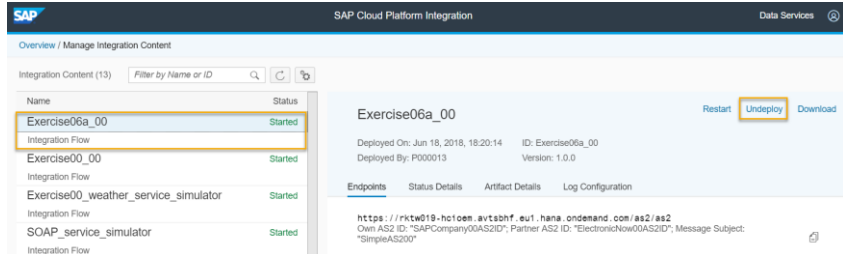
28) Check the generated file in the AS2 folder



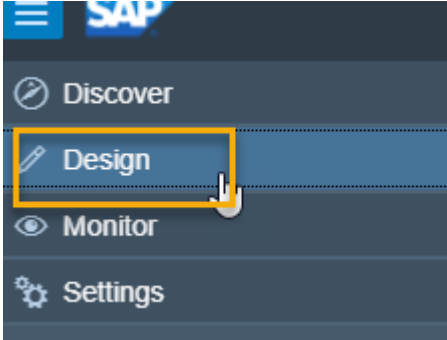
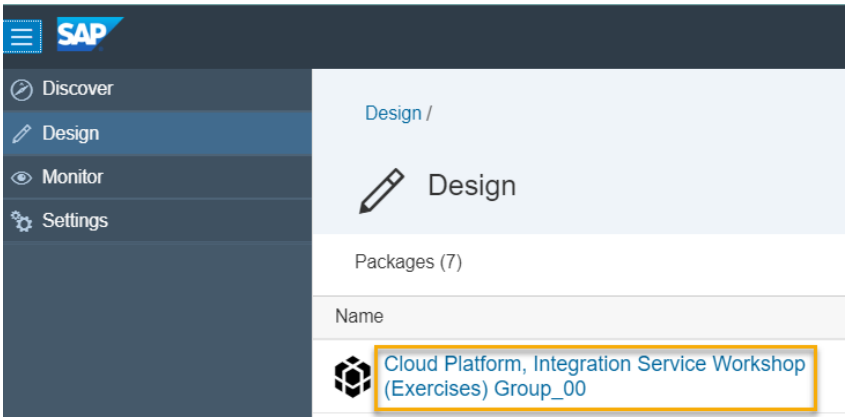

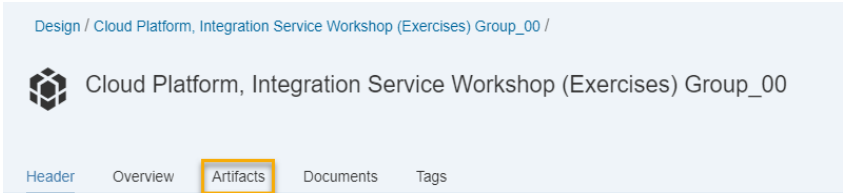
29) Finally, **Undeploy** the integration flow.

- Navigate to **Monitoring -> Manage Integration Content**
- Select **Exercise06a\_XX** integration flow.
- Click **Undeploy** button on the top right section.

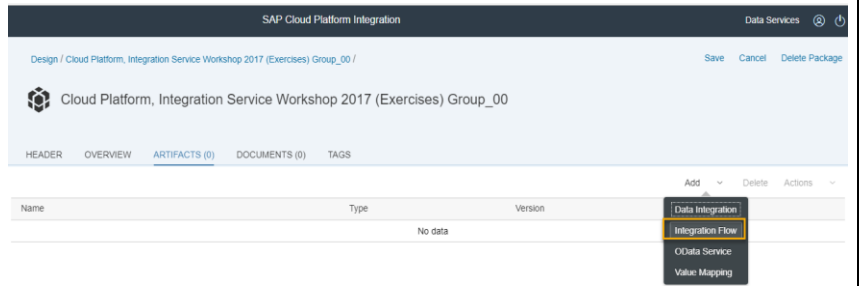
**Note:** This will help us to deploy the next part of this exercise without any issue. As next part of this exercise is having the same inbound message properties.



## EXERCISE 06B

Explanation	Screenshot
Follow these steps for integration flow creation in Web UI.	
1) Navigate to <b>Design</b> tab	
2) Select the package	
3) Click on <b>Edit</b>	
4) Navigate to <b>Artifacts</b> Tab	

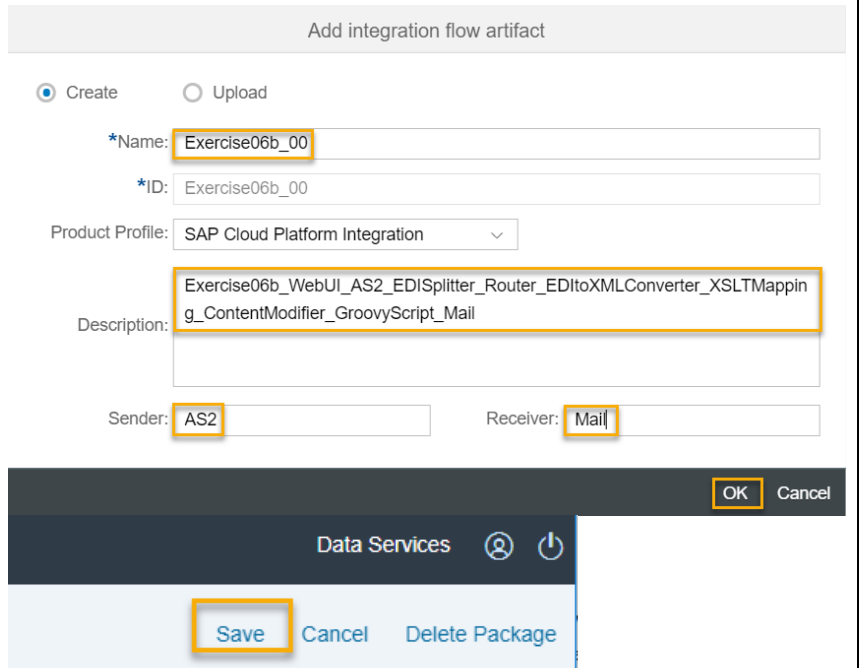
5) Click on **Add-> Integration Flow**



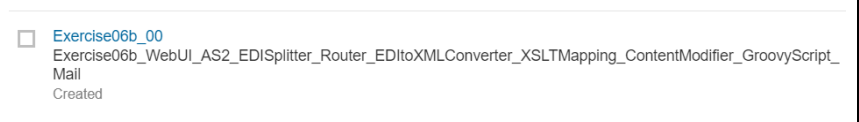
6) Select **Create** and enter following details:

- a. Name: **Exercise06b\_XX**
- b. Description: **Exercise06b\_WebUI\_AS2\_EDISplitter\_Router\_EDIttoXMLConverter\_XSLTMapping\_ContentModifier\_GroovyScript\_Mail**
- c. Sender: **AS2**
- d. Receiver: **Mail**
- e. Click on **OK**

**Note:** Please remember to save after every action



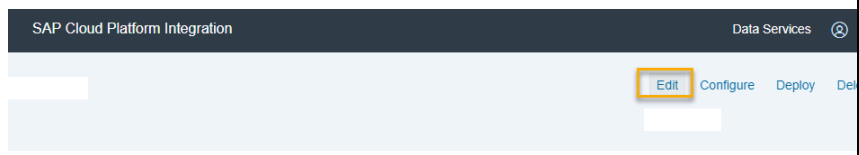
7) Click on **Exercise06b\_XX**

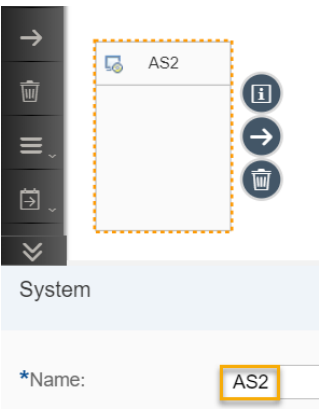
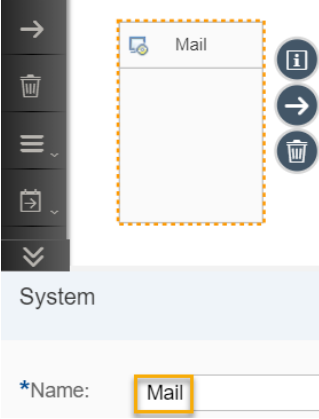
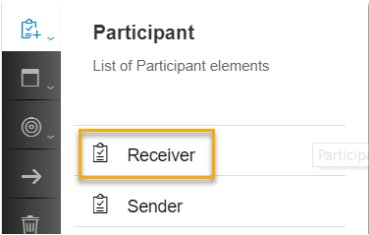
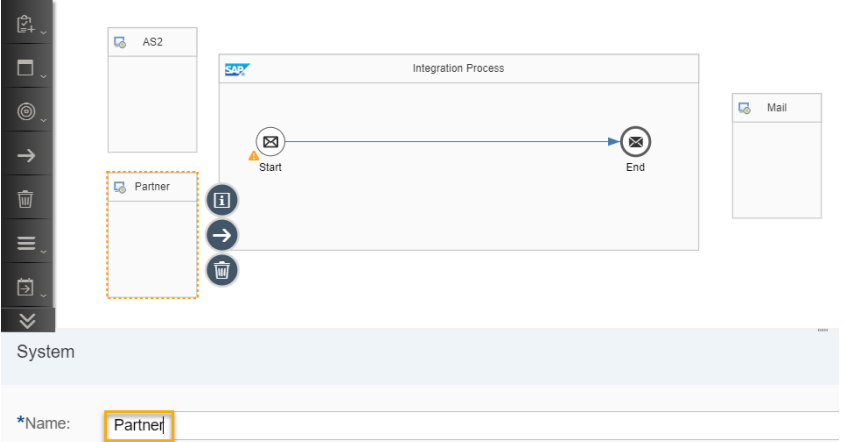


### Define and edit integration flow

### Follow steps to edit the integration flow

8) Click on **Edit** button on the upper corner on right hand side.



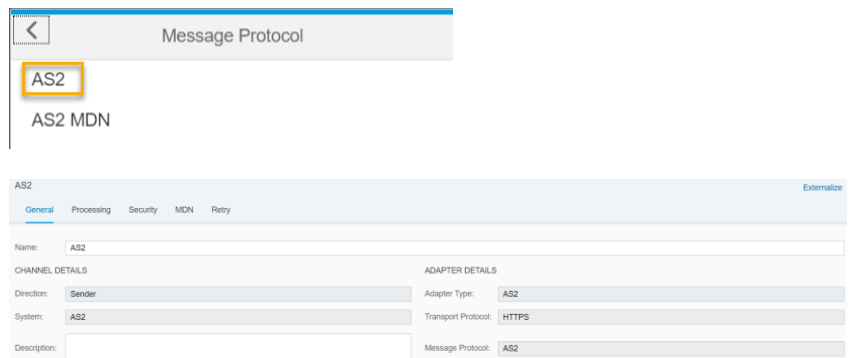
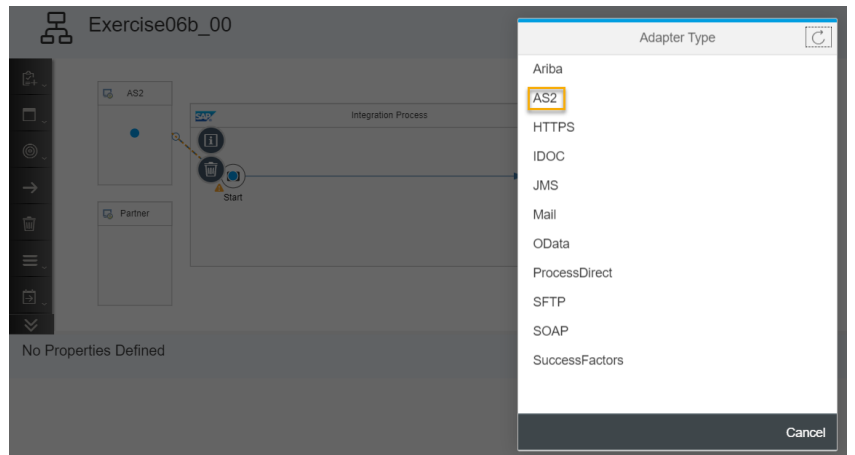
<p>9) Click on Sender system and set Name as <b>AS2</b></p>	
<p>10) Click on Receiver System and set Name as <b>Mail</b></p>	
<p>11) From the palette, select <b>Participants -&gt; Receiver</b> and drop it on iFlow editor window outside the Integration Process</p> <p>This receiver will route the 997 functional acknowledgments to the partner configured in the Mendelson platform</p>	
<p>12) Click on newly added Receiver system and set Name as <b>Partner</b></p>	



- 13) Click on **AS2** sender system, select **Connector** and drag it to **Start** message step.  
This would open the sender adapter list.

Select **AS2** adapter

Select **AS2** as Message Protocol



14) Switch to **Processing** tab, enter the following details:

- a. Message ID Left Part: \*
- b. Message ID Right Part: .\*
- c. Partner AS2 ID:  
**ElectronicNowXXAS2ID**
- d. Own AS2 ID:  
**SAPCompanyXXAS2ID**
- e. Message Subject:  
**SimpleAS2XX**

**Note:** Replace **XX** with your participant id

Save

## AS2

General **Processing** Security MDN Retry

### EXPECTED MESSAGES

\*Message ID Left Part: \*

\*Message ID Right Part: .\*

\*Partner AS2 ID: ElectronicNow00AS2ID

\*Own AS2 ID: SAPCompany00AS2ID

\*Message Subject: SimpleAS200

\*Number of Concurrent Processes: 1

\*Authorization: User Role

\*User Role: ESBMessaging.send

### MESSAGE SETTINGS

- ☐ Mandatory File Name
- ☐ Duplicate Message ID
- ☐ Duplicate File Name

15) Switch to **Security** tab, enter the following details:

- a. Decrypt Message: **checked**
- b. Private Key Alias: **picouser**
- c. Verify Signature: **checked**
- d. Public Key Alias: **key2**

## AS2

General Processing **Security** MDN Retry

### MESSAGE SECURITY

☒ Decrypt Message

\*Private Key Alias: picouser

☒ Verify Signature

\*Public Key Alias: key2

16) Switch to **MDN** tab, enter the following details:

- Private Key Alias for Signature: **picouser**
- Proxy Type: **On-Premise**
- Location ID: **LocXX**

**Note:** Replace **XX** with your participant id

Save

## AS2

General Processing Security **MDN** Retry

### RESPONSE MDN

Private Key Alias for Signature:

picouser

\*Signature Encoding:

binary

### ASYNCHRONOUS MDN CONNECTION SETTINGS

\*Proxy Type:

On-Premise

Location ID:

Loc00

\*Authentication:

None

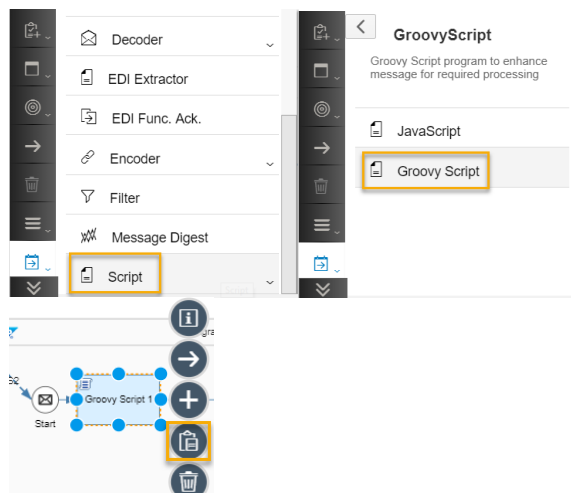
\*Timeout (in ms):

300000

17) A groovy script is provided which writes AS2 EDI messages in MPL which is quite useful for debugging purpose. It need to be added after **Start** message step in iFlow window using the following steps:

- From palette, select **Message Transformers -> Script -> Groovy Script** and drop it on the connection between **Start** message and **End** message in the integration flow. This would automatically create connections.
- Click on **Assign** icon.
- Upload **Log\_Payload.gsh** file from file system provided by the instructor.
- Rename the script as **Log AS2 Message**

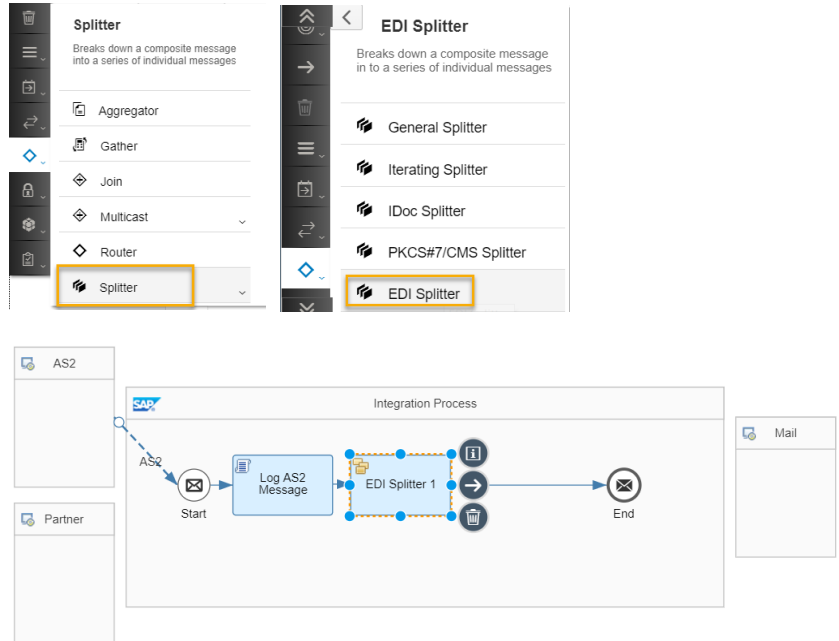
Save



### Log\_Payload.gsh

```
1 import com.sap.gateway.ip.core.customdev.util.Message;
2 import java.util.HashMap;
3
4 def Message processData(Message message) {
5     def body = message.getBody(java.lang.String) as String;
6     def messageLog = messageLogFactory.getMessageLog(message);
7     messageLog.addAttachmentAsString("Payload", body, "text/plain");
8     return message;
9 }
10 }
```

18) From the palette, select **Message Routing -> Splitter -> EDI Splitter** and drop it on the connection between **Groovy Script** and **End** message in the integration flow.



19) Switch to EDI Splitter **X12** tab, enter the following details:

- Validate Message:  
**Standard Validation**
- Transaction Mode:  
**Message**
- Create Acknowledgement:  
**Required**
- Interchange Number:  
**Use From EDI Message**

**EDI Splitter**

General Processing EDIFACT **X12**

\*Source Encoding: UTF-8

\*Validate Message: Standard Validation

\*Transaction Mode: Message

\*EDI Schema Definition: Integration Flow

\*Schemas:

☐ Schema Name

☐ Process Invalid Messages

\*Create Acknowledgement: Required

\*Interchange Number: Use From EDI Message

☐ Exclude AK3 and AK4

20) Under the same EDI Splitter **X12** tab, provide the **Schema** configuration by clicking on **Add** button

\*Schemas:

☐ Schema Name

☐  Select

**Add** Delete

Click on **Select -> Upload from File System** and browse for the **ASC-X12\_850\_004010.xsd** file which is available as a part of the exercise folder

You can see the uploaded XSD file under Schemas

Save

\*Schemas:

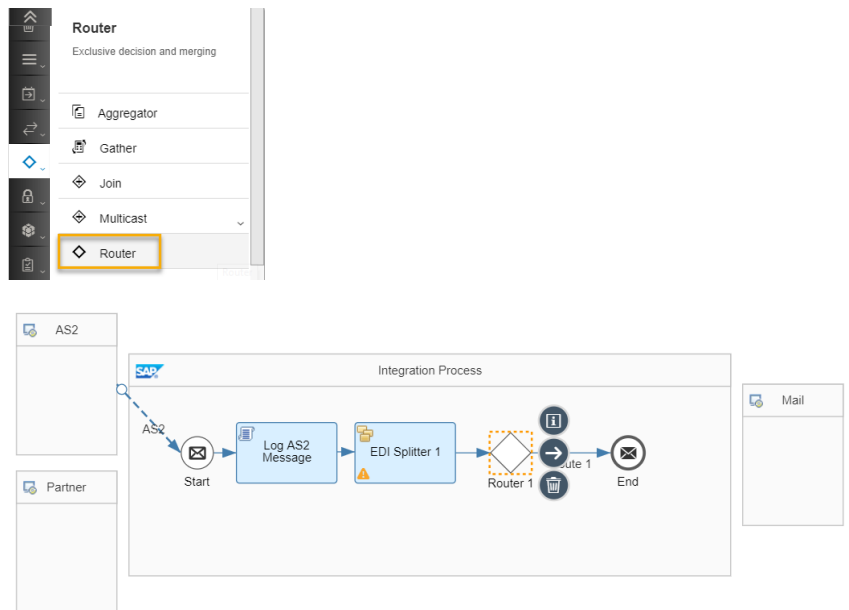
☐ Schema Name

☐ /xsd/ASC-X12\_850\_004010.xsd

☐ Process Invalid Messages

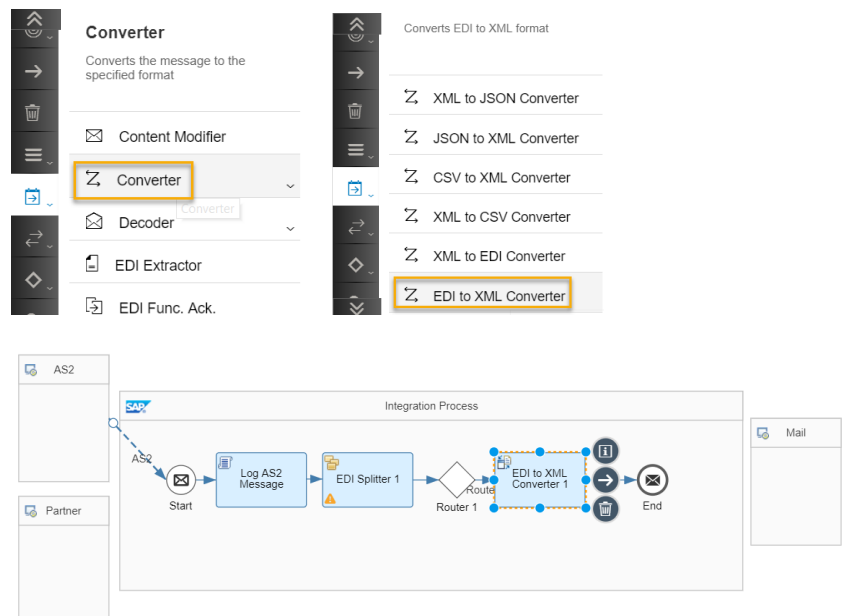
- 21) From the palette, select **Message Routing -> Router** and drop it on the connection between **EDI Splitter** and **End** message in the integration flow.

This is used to create a route for Acknowledgment and EDI valid messages.



- 22) From the palette, select **Message Transformers -> Converter -> EDI to XML Converter** and drop it on the connection between **Router** and **End** message in the integration flow.

This is used to transform EDI message to XML format.



- 23) Switch to EDI to XML Converter **X12** tab, provide the **Schema**

\*Schemas:

☐ Schema Name

☐

configuration by clicking on **Add** button.

Mainly we need to provide XSD for EDI message.

Click on **Select -> ASC-X12\_850\_004010** which is already referenced for EDI Splitter Schema mapping.

Save

**Note:** Ignore the errors on Integration Flow after save

Schema Name

Search

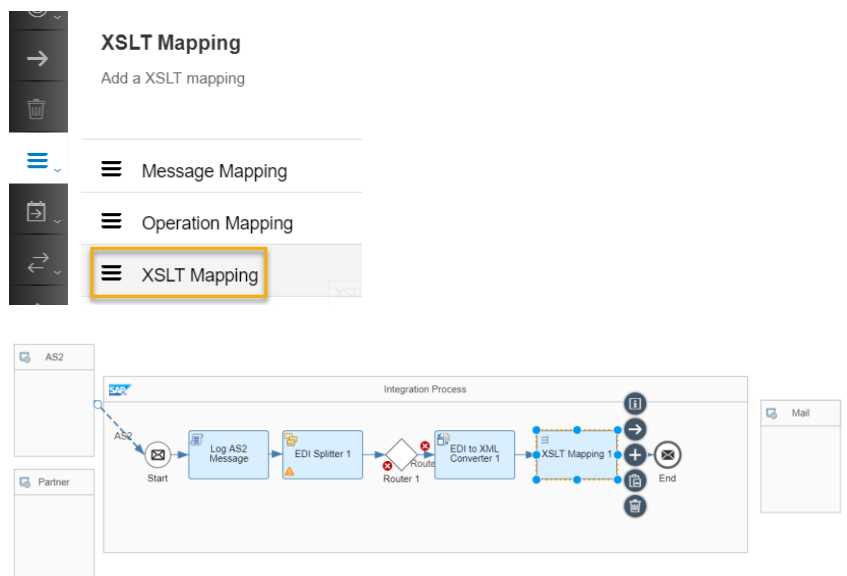
ASC-X12\_850\_004010

Upload from File System Cancel

\*Schemas:

Schema Name	Select
/xsd/ASC-X12_850_004010.xsd	Select

- 24) From the palette, select **Mapping -> XSLT Mapping** and drop it on the connection between **EDI to XML Converter** and **End** message in the integration flow.



- 25) Switch to **Processing** tab, provide the **Resource** configuration.

Click on **Select -> Upload from File System** and browse for the **XSLTMapping.xsl** file which is available as a part of the exercise folder

You can see the uploaded XSL file under Mappings

Save

XSLT Mapping

General Processing

\*Source: Integration Flow

\*Resource: <No resource assigned> Select

Output Format: Bytes

\*Source: Integration Flow

\*Resource: /mapping/XSLTMapping.xsl Select

Output Format: Bytes

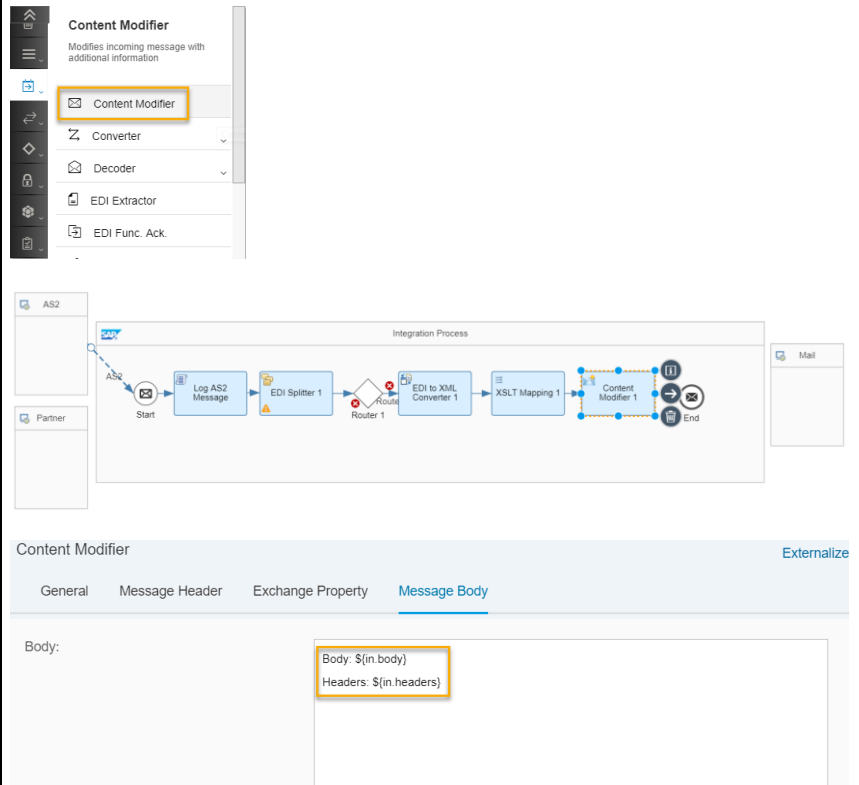
- 26) From the palette, select **Message Transformers -> Content Modifier** and drop it on the connection between **XSLT Mapping** and **End** message in the integration flow.

Switch to **Message Body** tab, enter the following text as Body:

**Body:**  
`${in.body}`

**Headers:**  
`${in.headers}`

Save



- 27) A groovy script is provided which writes IDOC Orders in MPL which is quite useful for debugging purpose. It need to be added after **Content Modifier** message step in iFlow window using the following steps:

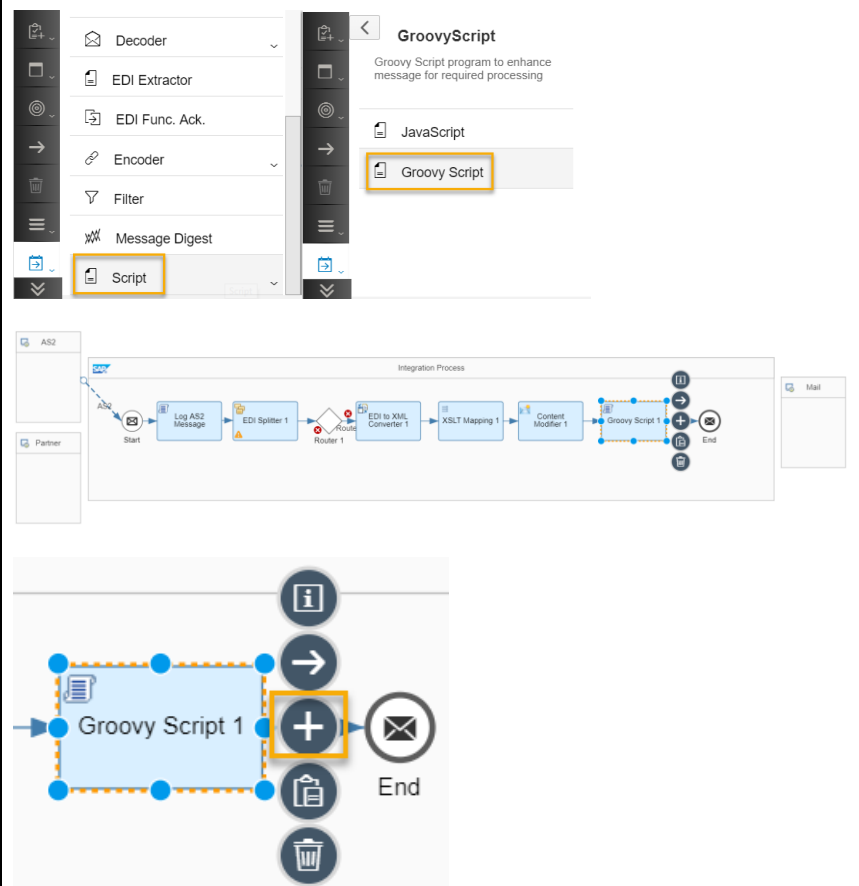
- From palette, select **Message Transformers -> Script -> Groovy Script** and drop it on the connection between **Content Modifier** and **End** message in the Integration flow.

- Click on **Create** icon

- Replace the code with following code:

```
import
com.sap.gateway.ip.core.customde
v.util.Message;
import java.util.HashMap;

def Message
processData(Message message) {
    def body =
message.getBody(java.lang.String)
as String;
    def messageLog =
messageLogFactory.getMessageLo
g(message);
```



```

        messageLog.addAttachmentAsString(" Log IDoc Orders",
        body, "text/plain");

        return message;
    }

```

- d. Click on **OK**
- e. Rename the script as **Log IDOC Orders**

Save


### script1.groovy

```

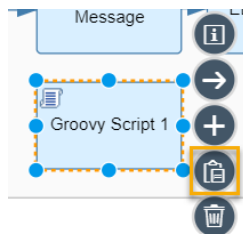
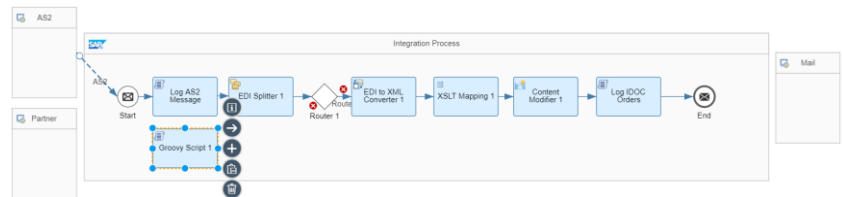
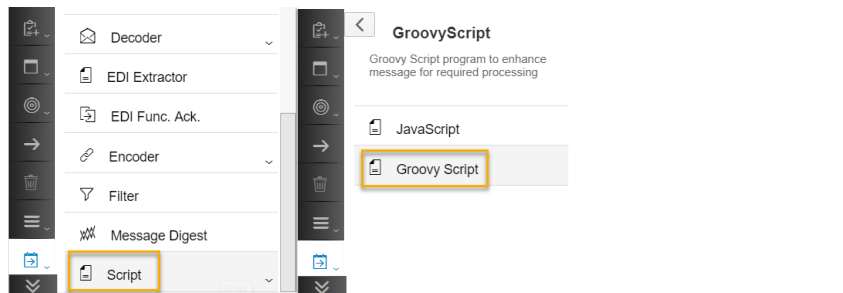
1  import com.sap.gateway.ip.core.customdev.util.Message;
2  import java.util.HashMap;
3
4  def Message processData(Message message) {
5      def body = message.getBody(java.lang.String) as String;
6      def messageLog = messageLogFactory.getMessageLog(message);
7      messageLog.addAttachmentAsString(" Log IDoc Orders", body, "text/plain");
8
9      return message;
10 }

```

28) A groovy script is provided which writes AS2 997 Acknowledgment messages in MPL which is quite useful for debugging purpose. It need to be added in Acknowledgment route which we will create.  
Add groovy script in iFlow window using the following steps:

- a. From palette, select **Message Transformers -> Script -> Groovy Script** and drop it in the integration flow below the **Log AS2 Message** groovy script flow step.
- b. Click on **Assign** icon .
- c. Upload **Log\_Functional\_Acknowledgement.gsh** file from file system provided by the instructor.
- d. Rename the script as **Log Functional Acknowledgement**

Save



### Log\_Functional\_Acknowledgement.gsh

```

1  import com.sap.gateway.ip.core.customdev.util.Message;
2  import java.util.HashMap;
3
4  def Message processData(Message message) {
5      def body = message.getBody(java.lang.String) as String;
6      def messageLog = messageLogFactory.getMessageLog(message);
7      messageLog.addAttachmentAsString("Log Functional Acknowledgement", body, "text/plain");
8
9      return message;
10 }

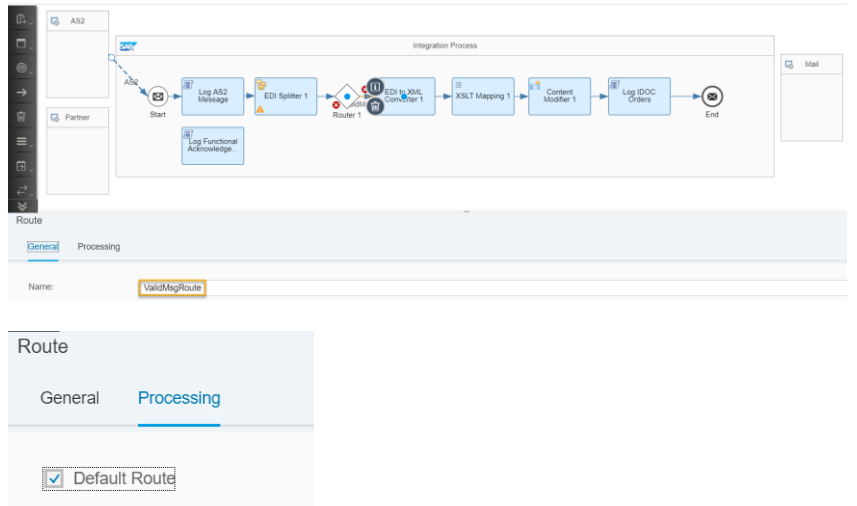
```



29) Select connection between **Router** and **EDI to XML Converter** and configure the route with following values:

- Name: **ValidMsgRoute**
- Check **Default Route**

Save

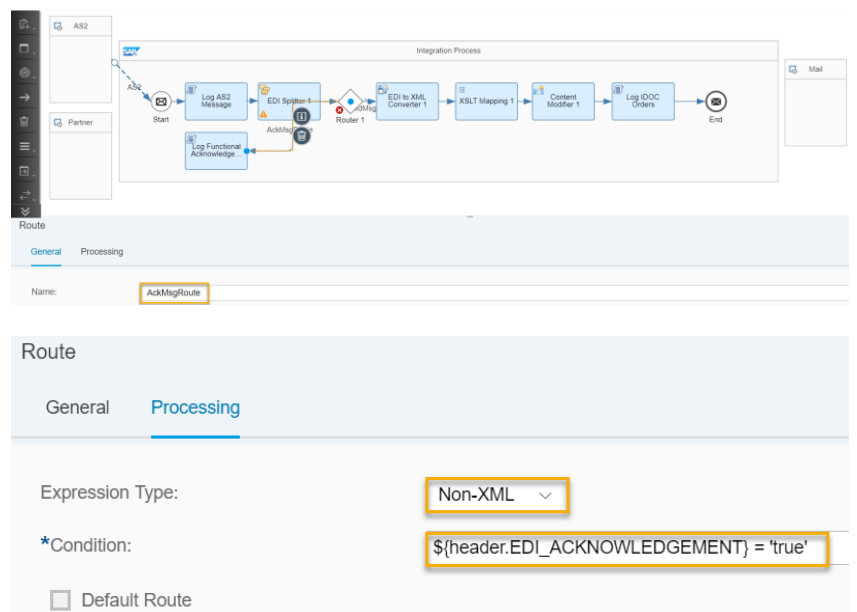


30) Now create an acknowledgement message route for EDI Splitter to send 997 functional acknowledgment to customer for processed EDI message by connecting Router's non- Default route to Log Functional Acknowledgment script.

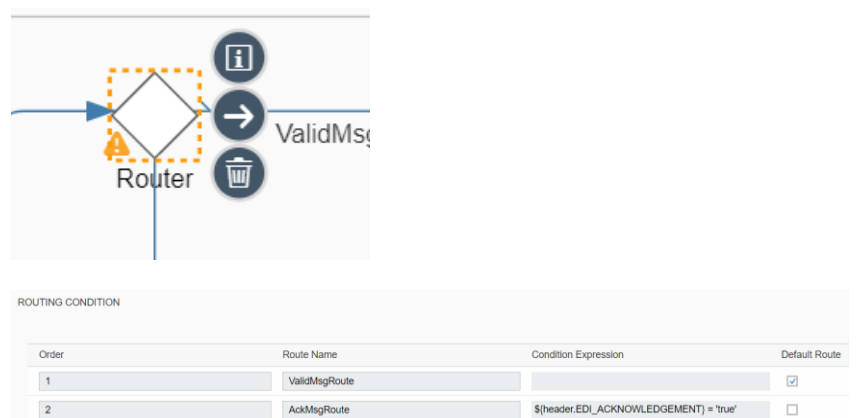
Draw the connection from **Router** to **Log Functional Acknowledgment** and configure the route with following values:

- Name: **AckMsgRoute**
- Expression Type: **Non-XML**
- Condition:  **$\$(header.EDI\_ACKNOWLEDGEMENT) = 'true'$**

Save

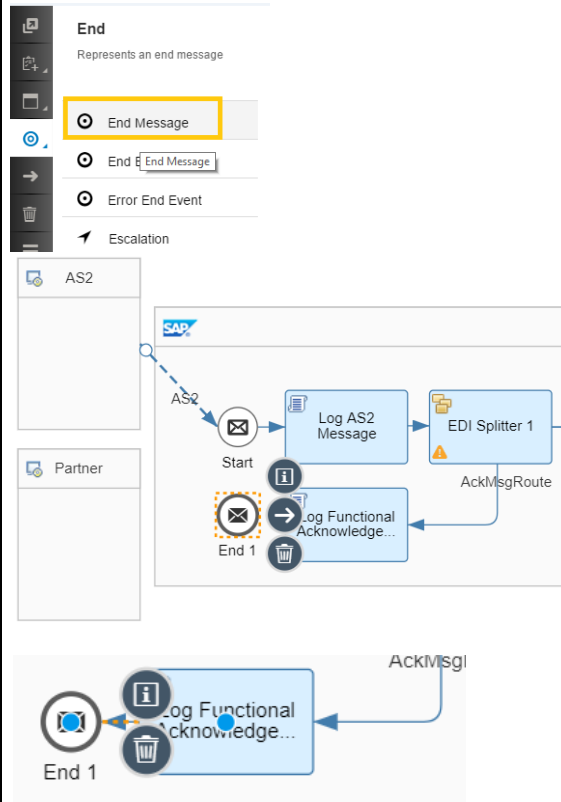


31) Now we can see two routing conditions for the router.



32) Click **Events**, select **End Message** and drop it on the integration flow

Connect **Log Functional Acknowledgment** groovy script to **End Message** event.



33) Click on **End 1** message step, select **Connector** and drag it to **Partner** receiver system. This would open the receiver adapter list.

Select **AS2** adapter



34) Switch to **Connection** tab, enter the following details:

- a. Recipient URL: **http://<Virtual Host>:<Virtual Port>/as2/HttpReceiver**  
E.g.  
<http://myhttp:8000/as2/HttpReceiver>
- b. Proxy Type: **On-Premise**
- c. Location ID: **LocXX**

**Note:** Here **Virtual Host** and **Virtual Port** are the virtual host and port addresses which you have entered in Cloud Connector.

Replace **XX** with your participant id

## AS2

General **Connection** Processing Security MDN

### RECEIPT INFORMATION

\*Recipient URL:

URL Parameters Pattern:

\*Proxy Type:

Location ID:

Authentication Type:

\*Timeout (in ms):

35) Switch to **Processing** tab, enter the following details:

- a. Own AS2 ID: **SAPCompanyXXAS2ID**
- b. Partner AS2 ID: **ElectronicNowXXAS2ID**
- c. Message Subject: **SendAck**
- d. Own E-mail address: **AS2@SAPCompany.com**
- e. Content-Type: **application/text**

**Note:** Replace **XX** with your participant id

Save

## AS2

General Connection **Processing** Security MDN

### MESSAGE INFORMATION

File Name:

Message ID Left Part:

Message ID Right Part:

\*Own AS2 ID:

\*Partner AS2 ID:

\*Message Subject:

\*Own E-mail address:

\*Content-Type:

Custom Headers Pattern:

\*Content Transfer Encoding:

36) Switch to **Security** tab, enter the following details:

- a. Compress Message: **checked**
- b. Sign Message: **checked**
- c. Private Key Alias: **picouser**
- d. Encrypt Message: **checked**
- e. Public Key Alias: **key2**

Save

AS2

General Connection Processing **Security** MDN

MESSAGE SECURITY

☒ Compress Message

☒ Sign Message

\*Algorithm:

SHA1

\*Private Key Alias:

picouser

☒ Encrypt Message

\*Algorithm:

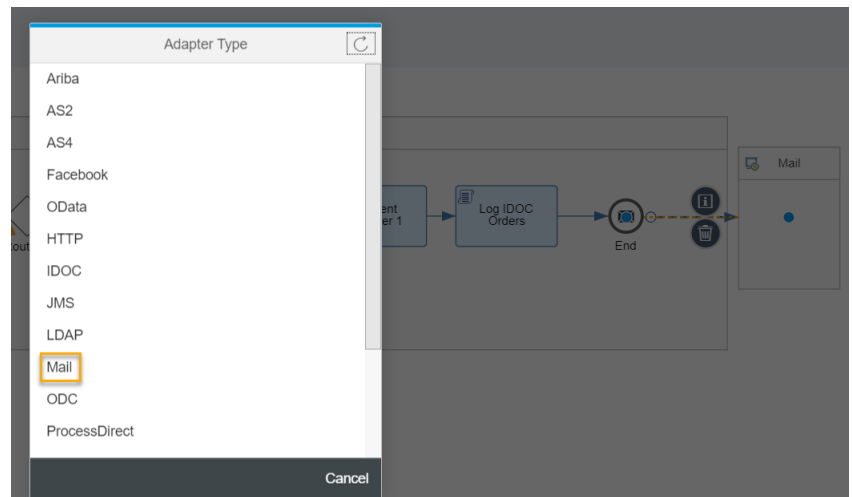
3DES

\*Public Key Alias:

key2

37) Click on **End** message step, select **Connector** and drag it to **Mail** receiver system. This would open the receiver adapter list.

Select **Mail** adapter



Mail

General Connection Security Externalize

Name: Mail

CHANNEL DETAILS

Direction: Receiver

System: Mail

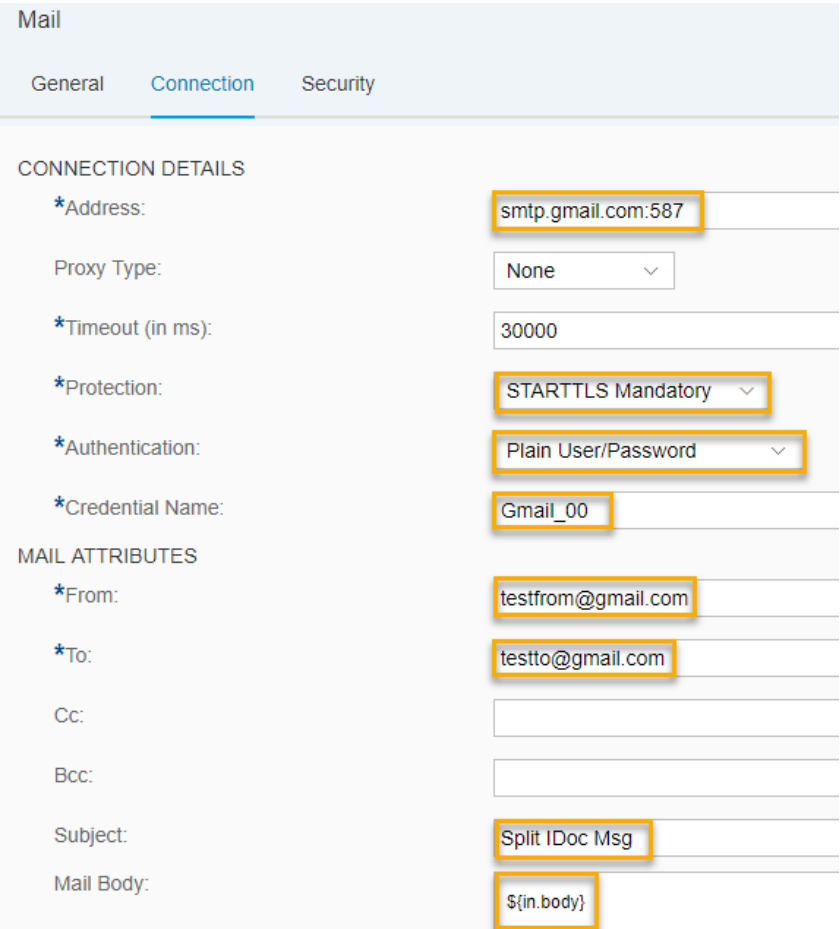
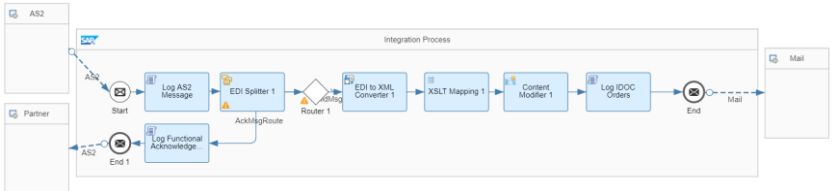

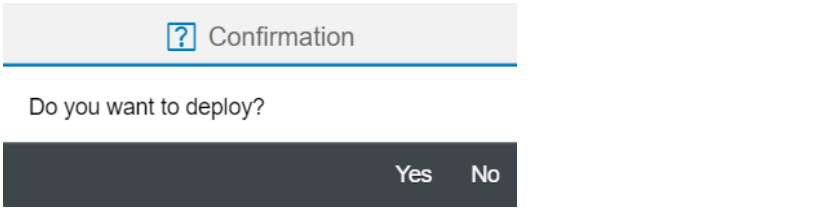
Description:

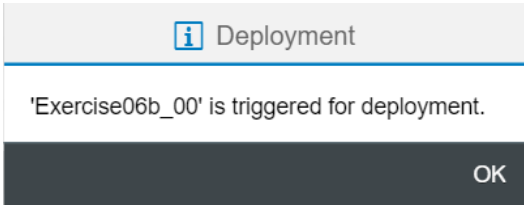
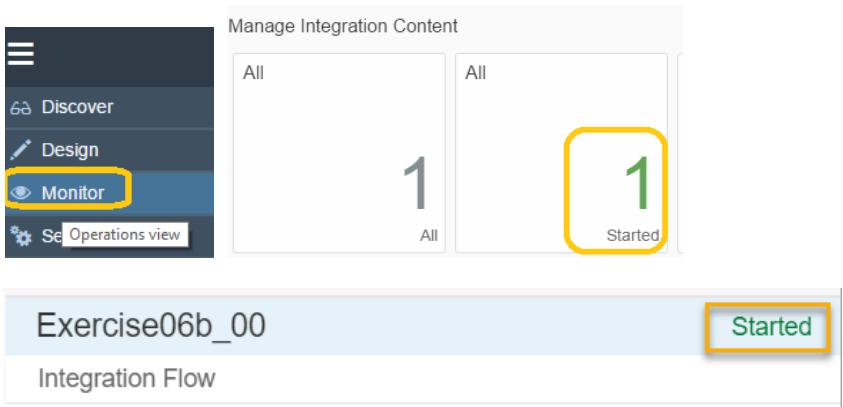
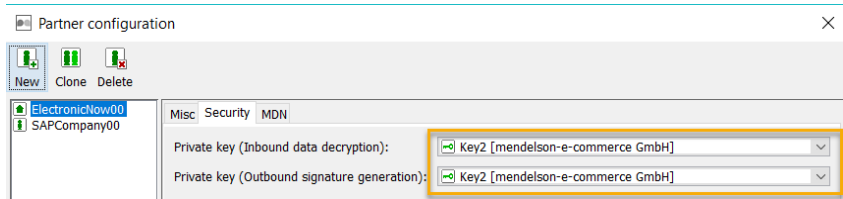
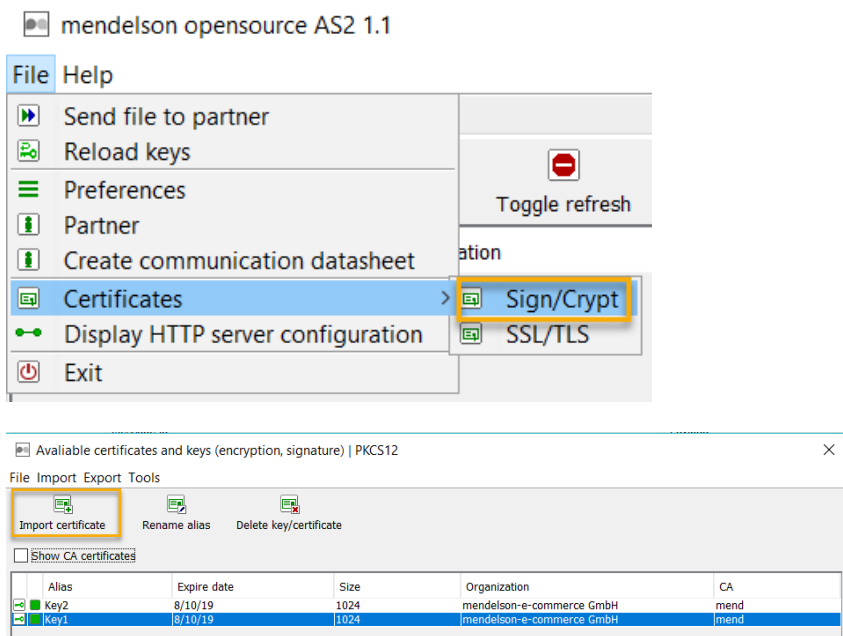
ADAPTER DETAILS

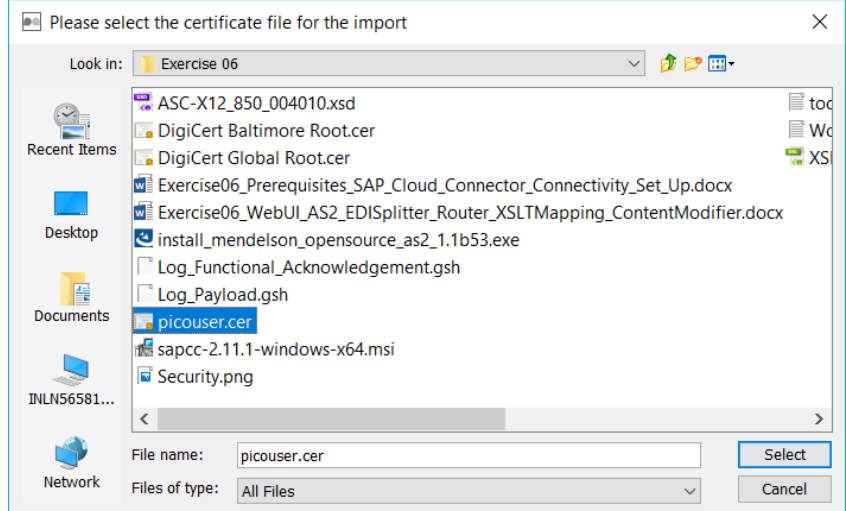
Adapter Type: Mail

Transport Protocol: SMTP

Message Protocol: None

<p>38) Switch to <b>Connection</b> tab, enter the following details:</p> <ol style="list-style-type: none"> <li>Address: <b>smtp.gmail.com:587</b></li> <li>Protection: <b>STARTTLS Mandatory</b></li> <li>Authentication: <b>Plain User/Password</b></li> <li>Credential Name: <b>Gmail_XX</b></li> <li>From: Mail ID from where you want to send the mail. (eg. <a href="mailto:testfrom@gmail.com">testfrom@gmail.com</a>)</li> <li>To: Mail ID where you want to receive the mail. (eg. <a href="mailto:testto@gmail.com">testto@gmail.com</a>)</li> <li>Subject: <b>Split IDoc Msg</b></li> </ol> <p>Save</p>	
<p>39) Your integration flow should look like this.</p>	
<p><b>Deploy Integration Project on tenant:</b></p>	<p><b>Follow this step to deploy Integration Project on tenant.</b></p>
<p>40) Press <b>Deploy</b> in the Integration Flow Task Bar.</p>	
<p>41) You will receive a confirmation. Click on <b>Yes</b>.</p> <p>Once the deployment is completed successfully you will receive a 2<sup>nd</sup> notification.</p>	

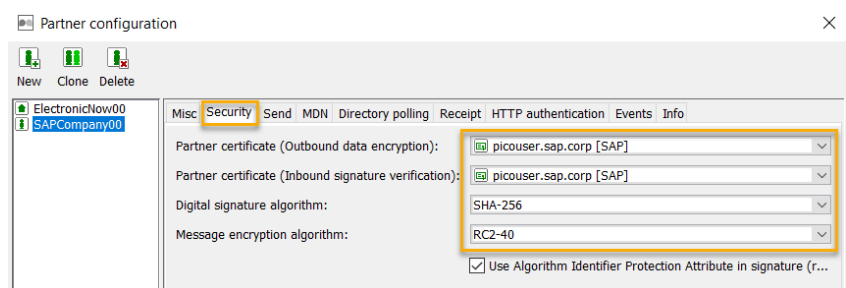
																
<p>42) Verify if the deployment is successful:</p> <p>In the first level Menu Bar switch to Section <b>Monitor</b> and then click on <b>All</b> in <b>Manage Integration Content</b>.</p> <p>You should see an entry with your integration flow.</p> <p>Check the 'status'. It should be in status <b>Started</b>.</p>																
<b>Execute end to end scenario:</b>	<b>Follow steps to execute end to end scenario.</b>															
<p>43) Modify the Mendelson AS2 local station Customer <b>ElectronicNowXX</b> security configuration and select <b>Key2</b> for both private key inbound/outbound security configuration.</p>																
<p>44) Import picouser certificate on to the Mandelson tool</p> <p>a. In the Mandelson tool navigate to <b>File -&gt; Certificates -&gt; Sign/Crypt</b></p> <p>b. Click on import certificate</p> <p>c. Click on Browse and from the Exercise folder, import the certificate <b>picouser.cer</b> file.</p> <p>This import the SAP cloud platform integration certificate used for encryption and sign validation</p>	 <table><thead><tr><th>Alias</th><th>Expire date</th><th>Size</th><th>Organization</th><th>CA</th></tr></thead><tbody><tr><td>Key2</td><td>8/10/19</td><td>1024</td><td>mendelson-e-commerce GmbH</td><td>mend</td></tr><tr><td>Key1</td><td>8/10/19</td><td>1024</td><td>mendelson-e-commerce GmbH</td><td>mend</td></tr></tbody></table>	Alias	Expire date	Size	Organization	CA	Key2	8/10/19	1024	mendelson-e-commerce GmbH	mend	Key1	8/10/19	1024	mendelson-e-commerce GmbH	mend
Alias	Expire date	Size	Organization	CA												
Key2	8/10/19	1024	mendelson-e-commerce GmbH	mend												
Key1	8/10/19	1024	mendelson-e-commerce GmbH	mend												



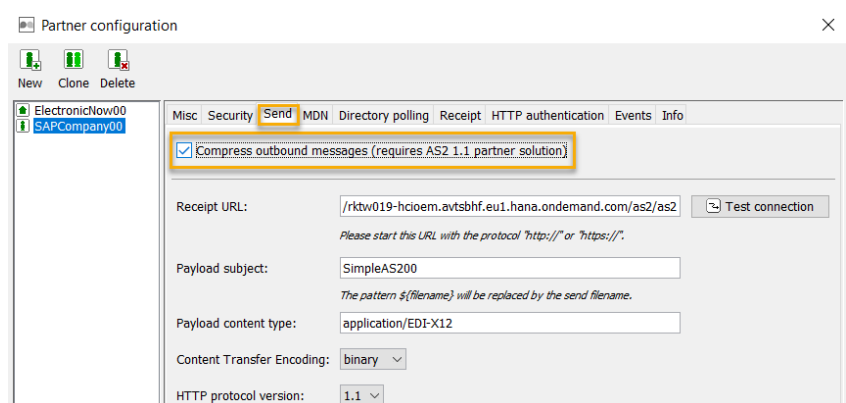
☐ Show CA certificates

	Alias	Expire date	Size
	Key2	8/10/19	1024
	Key1	8/10/19	1024
	picouser	5/17/19	2048

45) Choose the above imported certificate (picouser) for encryption and signature certificates and algorithms for SAPCompanyXX partner security.

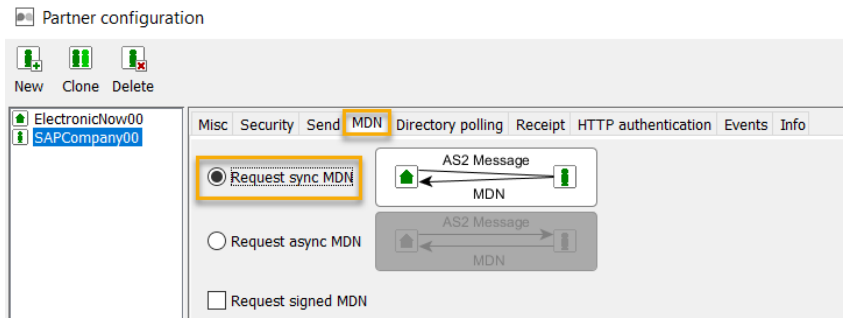


46) Select the Compress outbound messages in **Send** settings



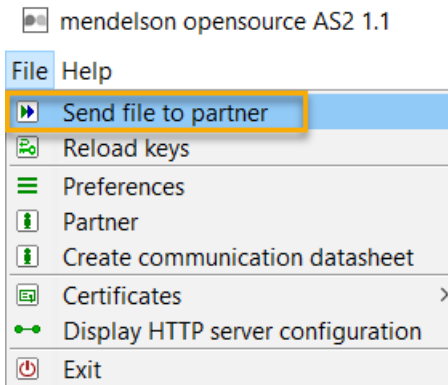
47) Request sync MDN option in MDN settings.

Click on **Ok**

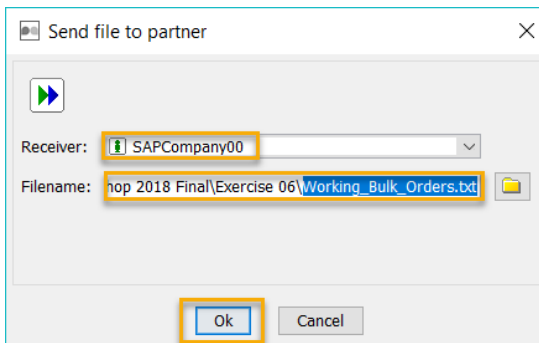


48) Let's test a simple AS2 message post from Mendelson to SAP Cloud Platform Integration Tenant

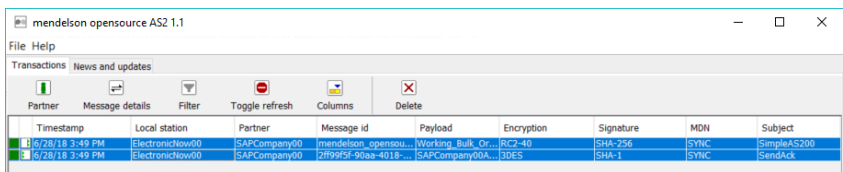
In the Mendelson tool, Choose **File -> Send file to partner**



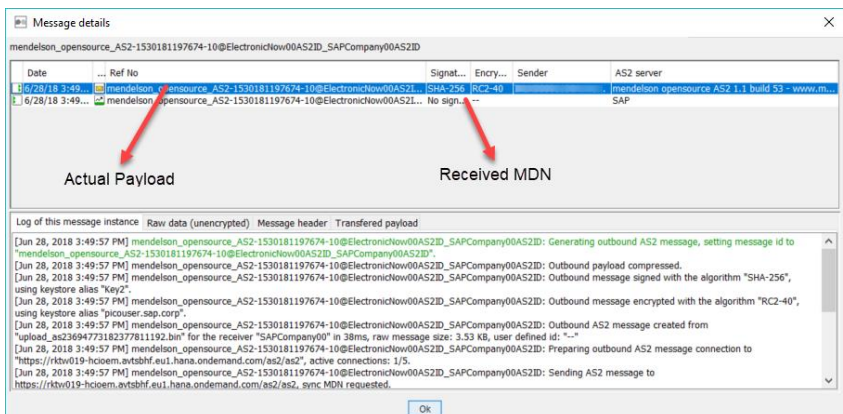
49) Choose **SAPCompanyXX** from the Receiver dropdown and select **Working\_Bulk\_Orders.txt** file and click Ok



50) You should see two successful messages in the Mandelson tool as shown below

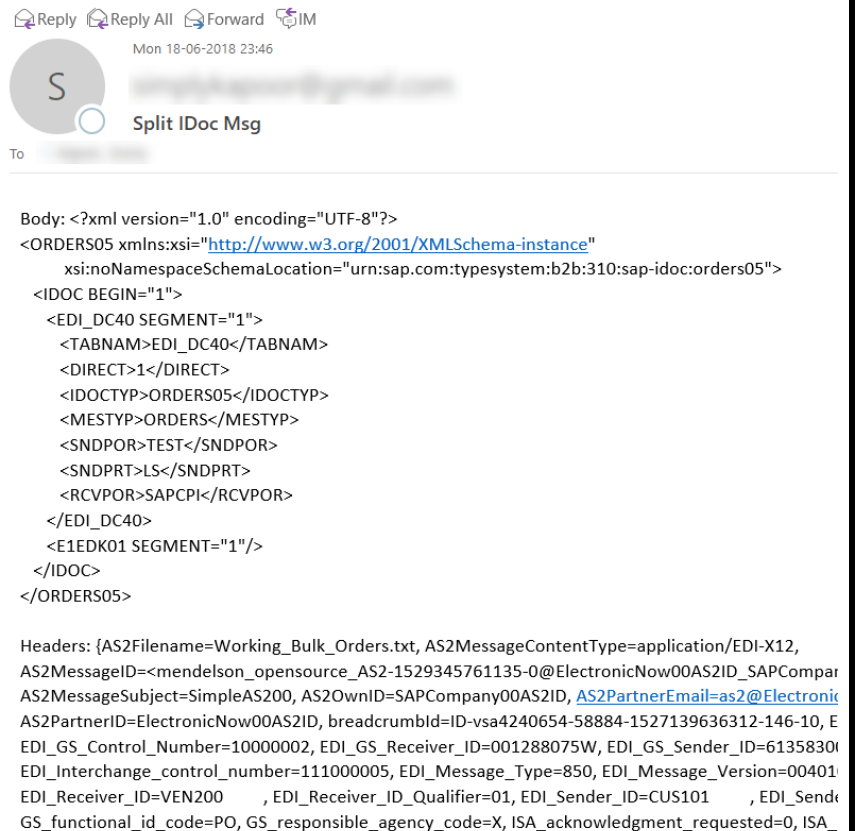


51) Double click on the message to see the message that was sent and the MDN received back





52) Check the Mail, you should receive a mail of IDOC SALES Orders in the configured mail account



Reply Reply All Forward IM  
Mon 18-06-2018 23:46

**Split IDoc Msg**

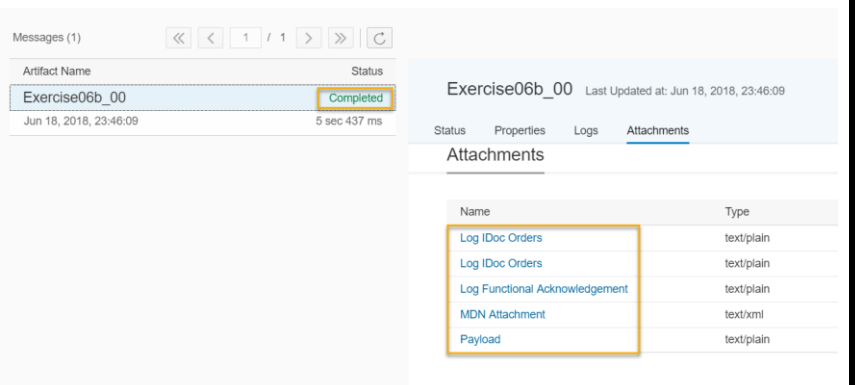
To: [redacted]

Body: <?xml version="1.0" encoding="UTF-8"?>  
<ORDERS05 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:noNamespaceSchemaLocation="urn:sap.com:typesystem:b2b:310:sap-idoc:orders05">  
<IDOC BEGIN="1">  
<EDI\_DC40 SEGMENT="1">  
<TABNAM>EDI\_DC40</TABNAM>  
<DIRECT>1</DIRECT>  
<IDOC TYP>ORDERS05</IDOC TYP>  
<MESTYP>ORDERS</MESTYP>  
<SNDPDR>TEST</SNDPDR>  
<SNDPRT>LS</SNDPRT>  
<RCVPOR>SAPCI</RCVPOR>  
</EDI\_DC40>  
<E1EDK01 SEGMENT="1"/>  
</IDOC>  
</ORDERS05>

Headers: {AS2Filename=Working\_Bulk\_Orders.txt, AS2MessageContentType=application/EDI-X12, AS2MessageID=<mendelson\_opensource\_AS2-1529345761135-0@ElectronicNow00AS2ID\_SAPComparAS2MessageSubject=SimpleAS200, AS2OwnID=SAPCompany00AS2ID, AS2PartnerEmail=as2@ElectronicAS2PartnerID=ElectronicNow00AS2ID, breadcrumbId=ID-vsa4240654-58884-1527139636312-146-10, EEDI\_GS\_Control\_Number=10000002, EDI\_GS\_Receiver\_ID=001288075W, EDI\_GS\_Sender\_ID=61358301EDI\_Interchange\_control\_number=111000005, EDI\_Message\_Type=850, EDI\_Message\_Version=00401EDI\_Receiver\_ID=VEN200, EDI\_Receiver\_ID\_Qualifier=01, EDI\_Sender\_ID=CUS101, EDI\_Sender\_GS\_functional\_id\_code=PO, GS\_responsible\_agency\_code=X, ISA\_acknowledgment\_requested=0, ISA\_

53) On the CPI tenant, Navigate to **Monitor Message processing** to see a Completed message with an MPL attachments.

**Note:** you can validate Logs for information on EDI messages being sent, IDOC Sales Orders created and Functional Acknowledgments delivered to partner



Messages (1) << < 1 / 1 > >> Refresh

Artifact Name	Status
Exercise06b_00	Completed

Jun 18, 2018, 23:46:09 5 sec 437 ms

Exercise06b\_00 Last Updated at: Jun 18, 2018, 23:46:09

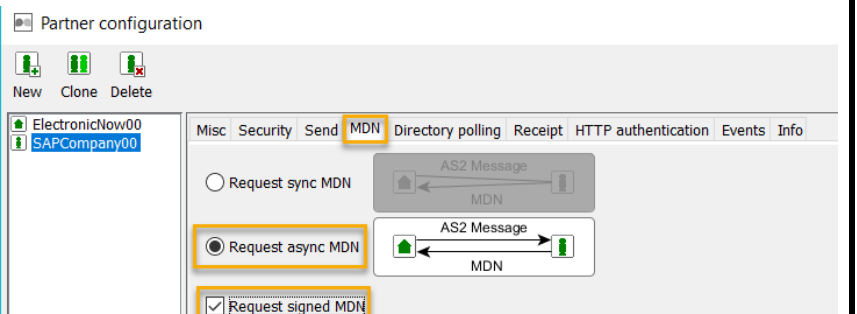
Status Properties Logs Attachments

Attachments

Name	Type
Log IDoc Orders	text/plain
Log IDoc Orders	text/plain
Log Functional Acknowledgement	text/plain
MDN Attachment	text/xml
Payload	text/plain

54) Request async MDN option in **MDN** settings.

Click on **Ok**



Partner configuration

New Clone Delete

ElectronicNow00  
SAPCompany00

Misc Security Send MDN Directory polling Receipt HTTP authentication Events Info

☐ Request sync MDN

☒ Request async MDN

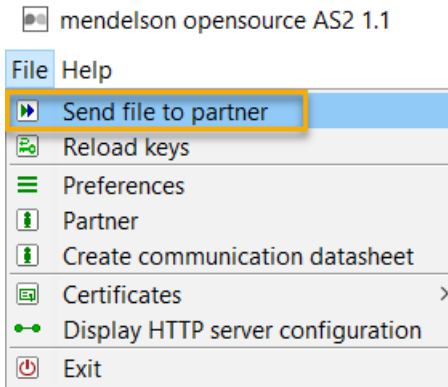
☒ Request signed MDN

AS2 Message  
MDN

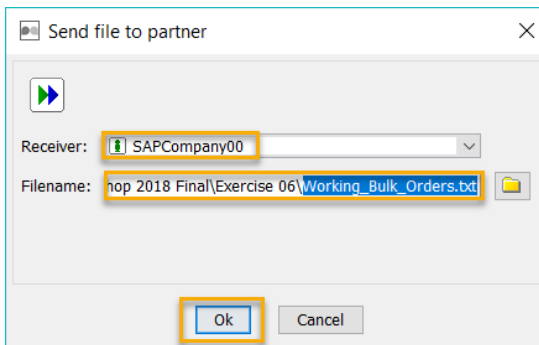
AS2 Message  
MDN

55) Let's test a simple AS2 message post from Mendelson to SAP Cloud Platform Integration Tenant

In the Mendelson tool, Choose **File -> Send file to partner**

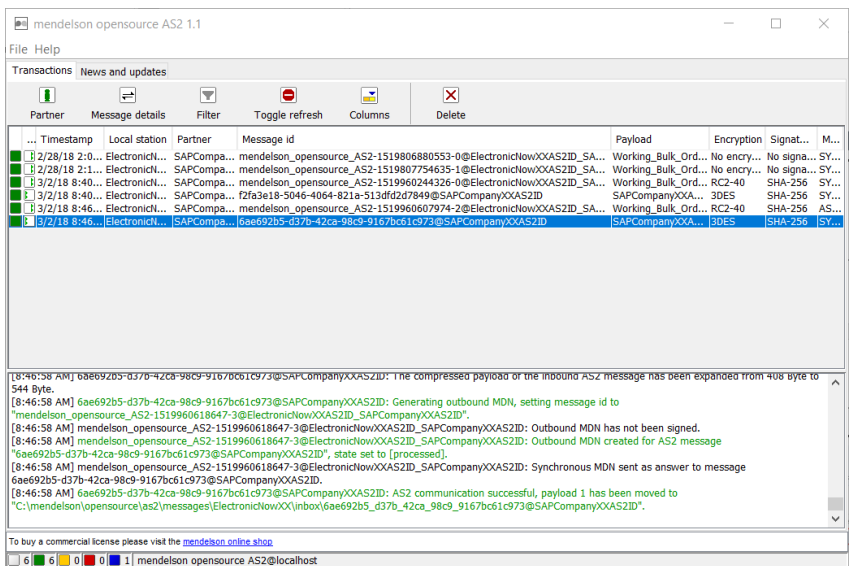


56) Choose **SAPCompanyXX** from the Receiver dropdown and select **Working\_Bulk\_Orders.txt** file and click Ok



57) You should see a successful message in the Mandelson tool as shown below.

**Note:** Message in green indicates, successful post



58) Check the Mail, you should receive a mail of IDOC SALES Orders in the configured mail account in Asynchronous MDN use case as well.

Reply Reply All Forward IM  
Mon 18-06-2018 23:46

S  
IDoc Msg

To

Body: <?xml version="1.0" encoding="UTF-8"?>  
<ORDERS05 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:noNamespaceSchemaLocation="urn:sap.com:typesystem:b2b:310:sap-idoc:orders05">  
<IDOC BEGIN="1">  
<EDI\_DC40 SEGMENT="1">  
<TABNAM>EDI\_DC40</TABNAM>  
<DIRECT>1</DIRECT>  
<IDDOCTYP>ORDERS05</IDDOCTYP>  
<MESTYP>ORDERS</MESTYP>  
<SNDPOR>TEST</SNDPOR>  
<SNDPRT>LS</SNDPRT>  
<RCVPOR>SAPCPI</RCVPOR>  
</EDI\_DC40>  
<E1EDK01 SEGMENT="1"/>  
</IDOC>  
</ORDERS05>

Headers: {AS2Filename=Working\_Bulk\_Orders.txt, AS2MessageContentType=application/EDI-X12, AS2MessageID=<mendelson\_opensource\_AS2-1529345761135-0@ElectronicNow00AS2ID\_SAPCompar AS2MessageSubject=SimpleAS200, AS2OwnID=SAPCompany00AS2ID, AS2PartnerEmail=as2@Electronic AS2PartnerID=ElectronicNow00AS2ID, breadcrumbId=ID-vsa4240654-58884-1527139636312-146-10, E EDI\_GS\_Control\_Number=10000002, EDI\_GS\_Receiver\_ID=001288075W, EDI\_GS\_Sender\_ID=6135830 EDI\_Interchange\_control\_number=111000005, EDI\_Message\_Type=850, EDI\_Message\_Version=00401 EDI\_Receiver\_ID=VEN200 , EDI\_Receiver\_ID\_Qualifier=01, EDI\_Sender\_ID=CUS101 , EDI\_Sender GS\_functional\_id\_code=PO, GS\_responsible\_agency\_code=X, ISA\_acknowledgment\_requested=0, ISA\_

59) On the CPI tenant, Navigate to **Monitor Message processing** to see a Completed message with an MPL attachments.

**Note:** you can validate Logs for information on EDI messages being sent, IDOC Sales Orders created and Functional Acknowledgments delivered to partner

Messages (1) << < 1 / 1 > >> ↺

Artifact Name	Status
Exercise06b_00	Completed
Jun 18, 2018, 23:46:09	5 sec 437 ms

Exercise06b\_00 Last Updated at: Jun 18, 2018, 23:46:09

Status Properties Logs Attachments

Attachments

Name	Type
Log IDoc Orders	text/plain
Log IDoc Orders	text/plain
Log Functional Acknowledgement	text/plain
MDN Attachment	text/xml
Payload	text/plain

