TECH BLOG

Back To Top

Menu

September 30, 2020 Sudarshan Patwardhan

SAP

Message Mapping and Value Mapping in SAP CPI

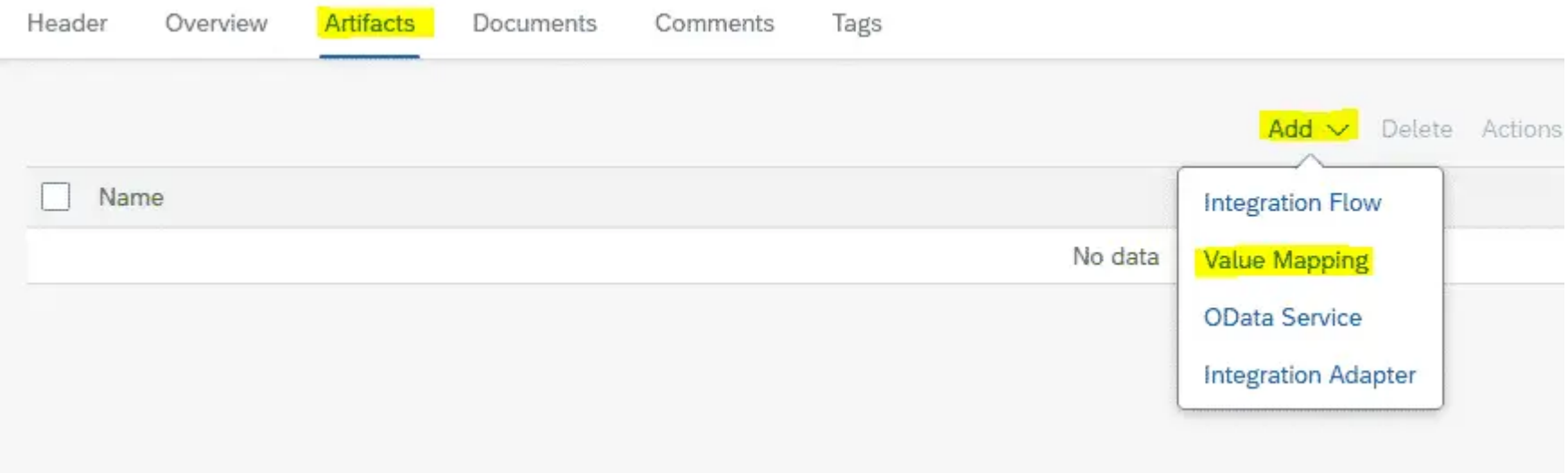
Let's have a look at what is Message mapping in SAP CPI, In CPI message mapping is required the type of message is different on your sender and receiver systems. In SAP PO message mapping we call this has Graphical mapping or Java Mapping according to the way we do this. This article will help you understand the Implementation of Message Mapping and Value Mapping in SAP CPI.

Table of Contents

Back To Top

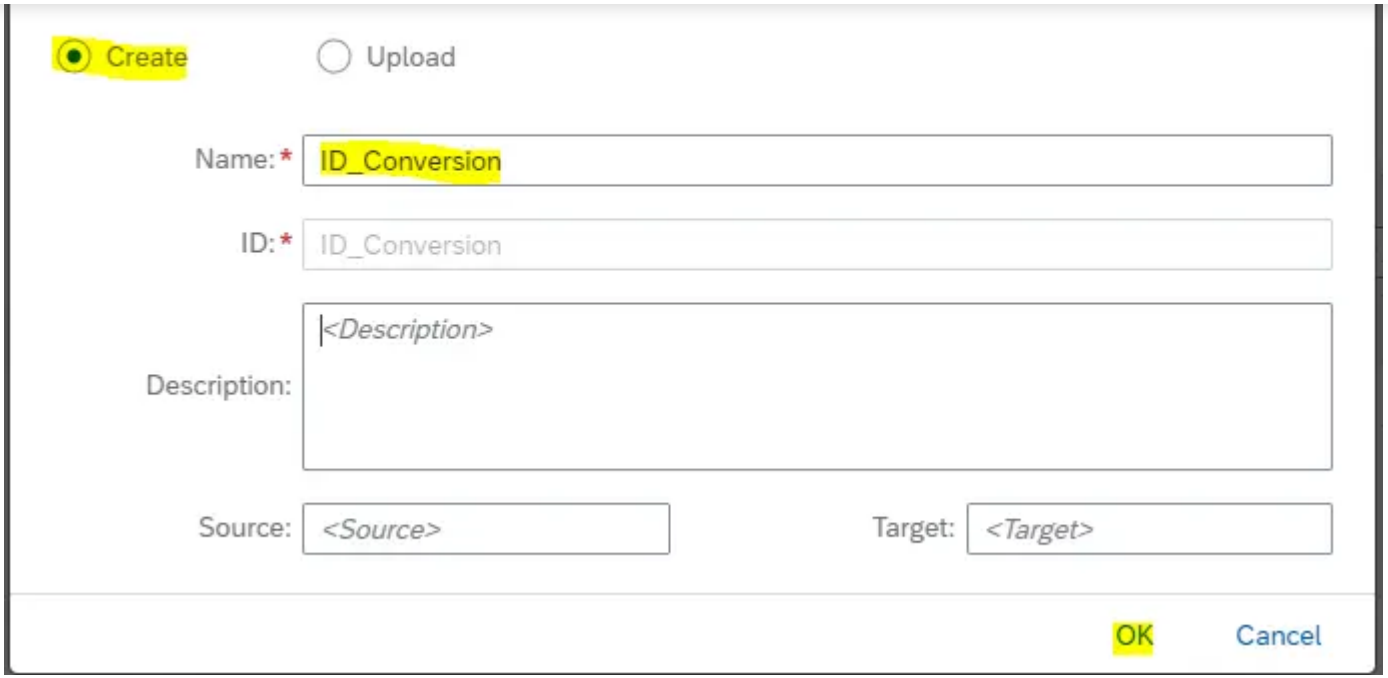
CREATING SAP CPI VALUE MAPPING

Step 1 : Go to **Artifacts** tab of your Package. **Add –> Value Mapping**, This will add sap cpi value mapping.

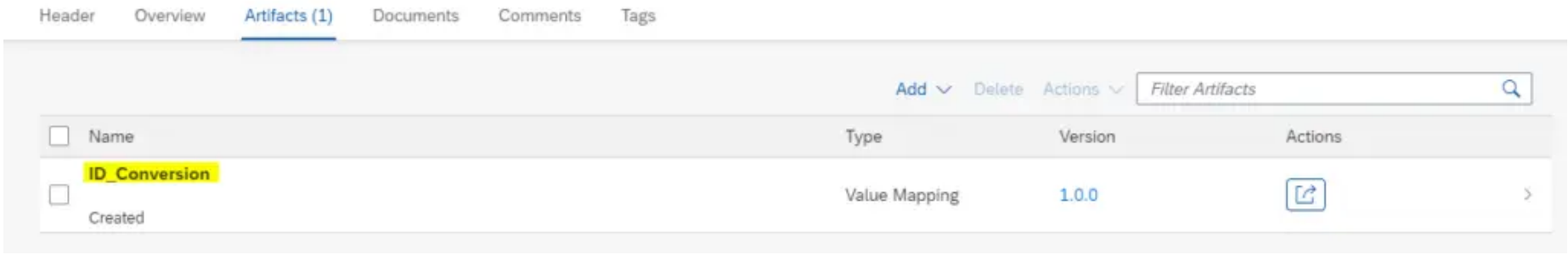


Step 2 : Select **Create** Radio Button and Give a **valid Name**. (This step is sap cpi namespace)

Back To Top

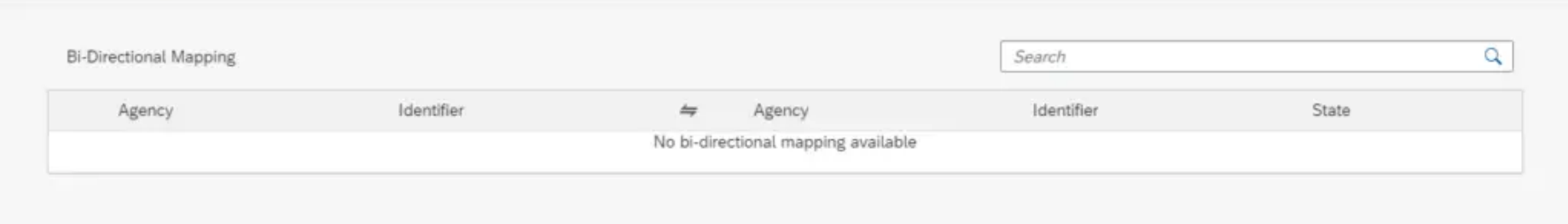


Step 3 : **Click** on the new Value Mapping Created.

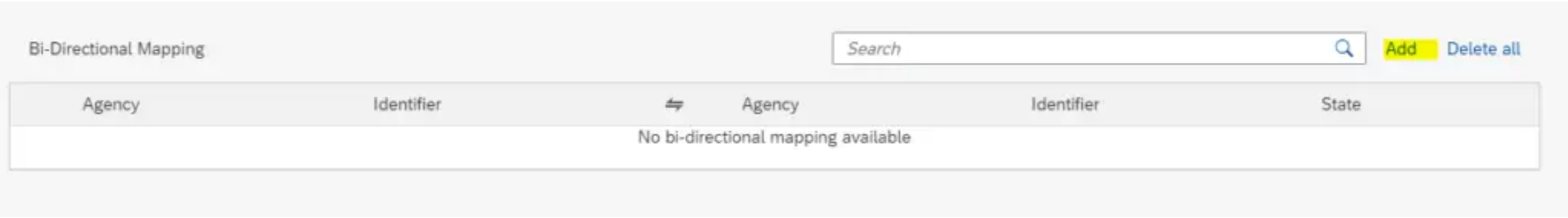


Step 4 : Click on **edit**

Back To Top



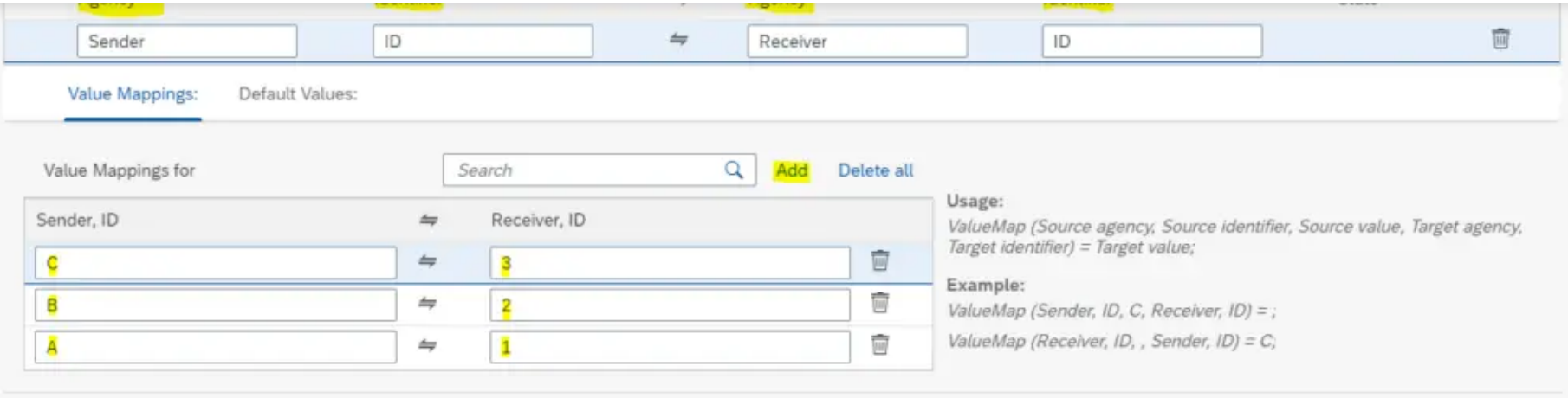
Step 5 : Click on **Add**.



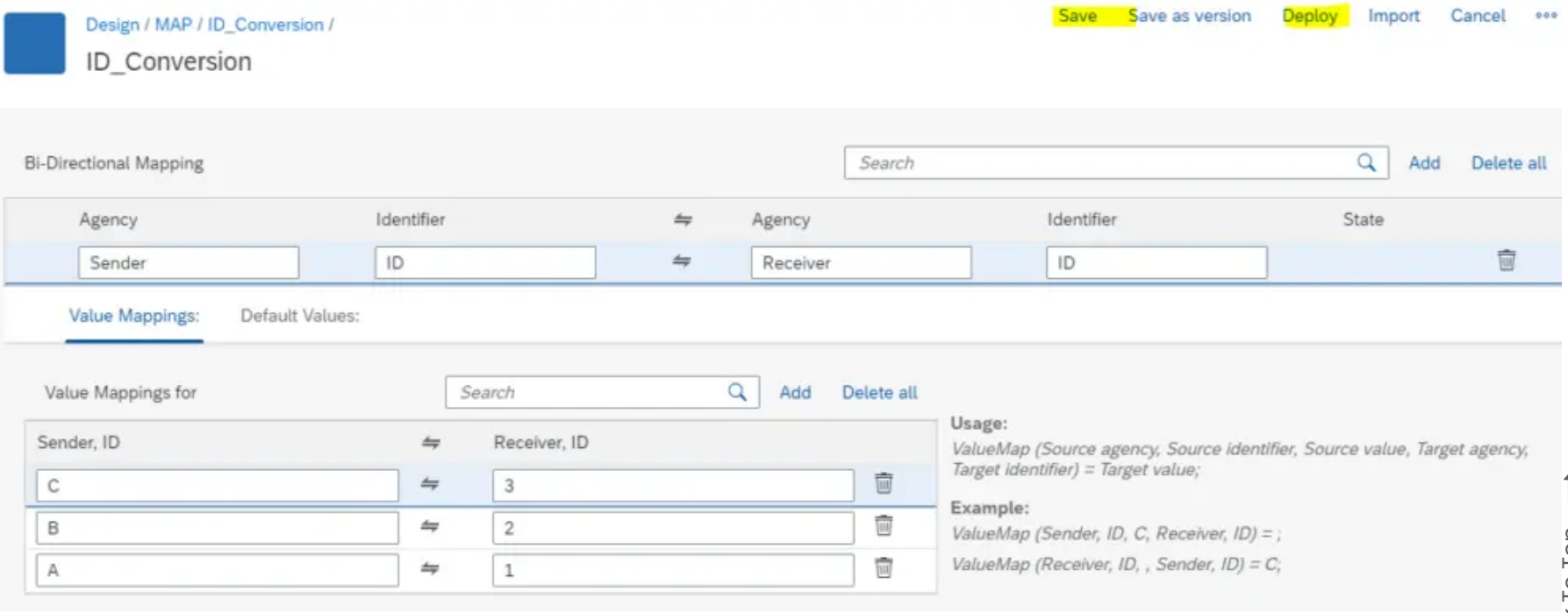
Step 6 : Fill the required details and Conversion Values.

Source	Target
A	1
B	2
C	3

Back To Top

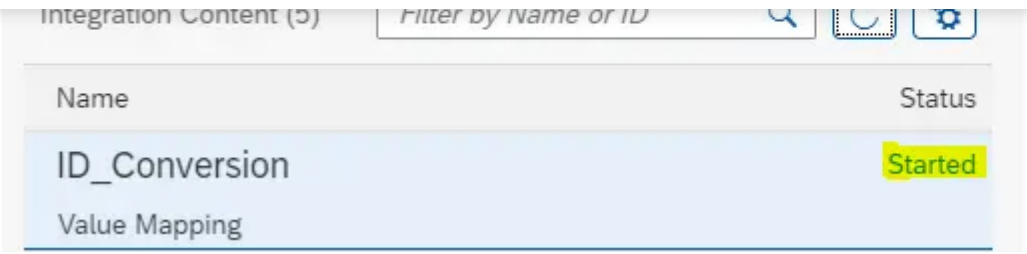


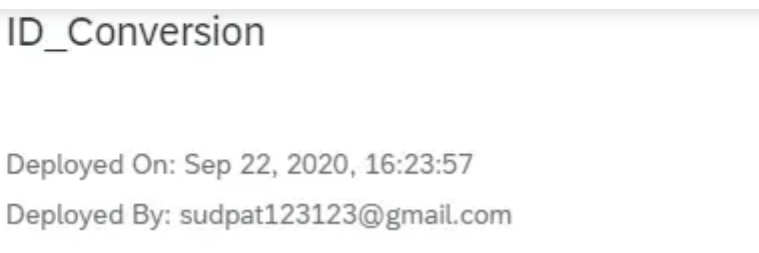
Step 7 : **Save** and **Deploy**.



Step 8 : In **Overview** Window, Make sure that the Value Mapping is **started**.

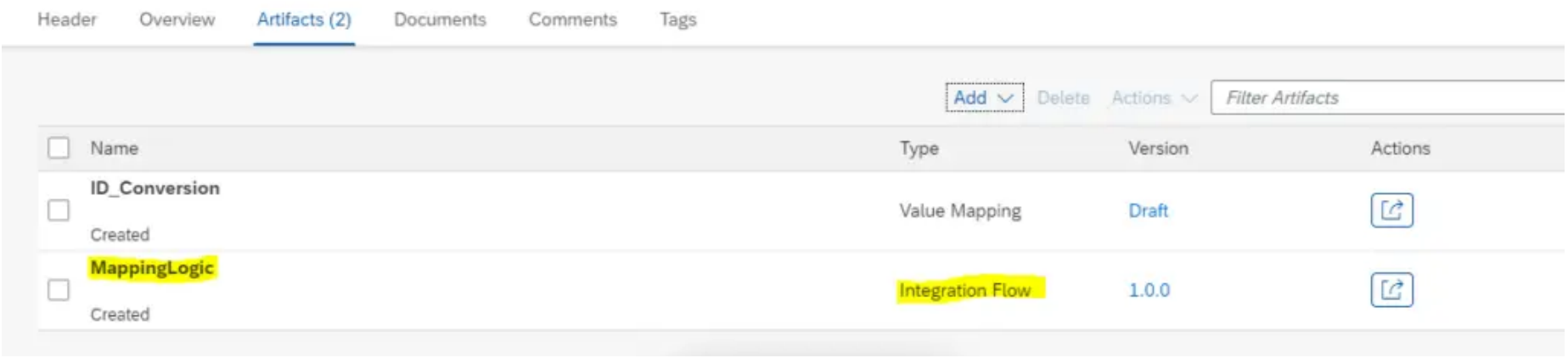
Back To Top





CREATING INTEGRATION FLOW

Step 9 : Go to **Design** Window and **Create** an **Integration Flow** under Artifacts.

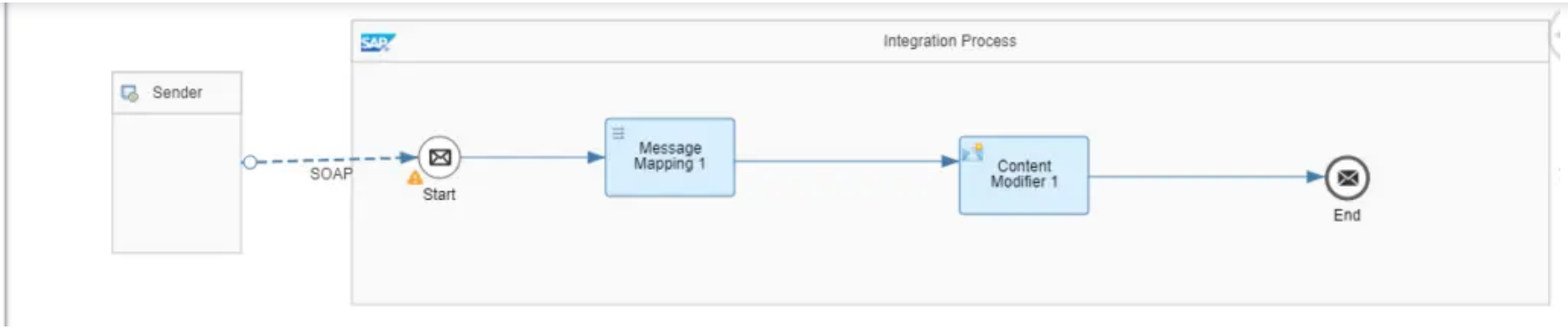


Step 10 : **Open** the Integration Flow in **edit** Mode. Add a ***Message Mapping*** and a ***Content Modifier***.

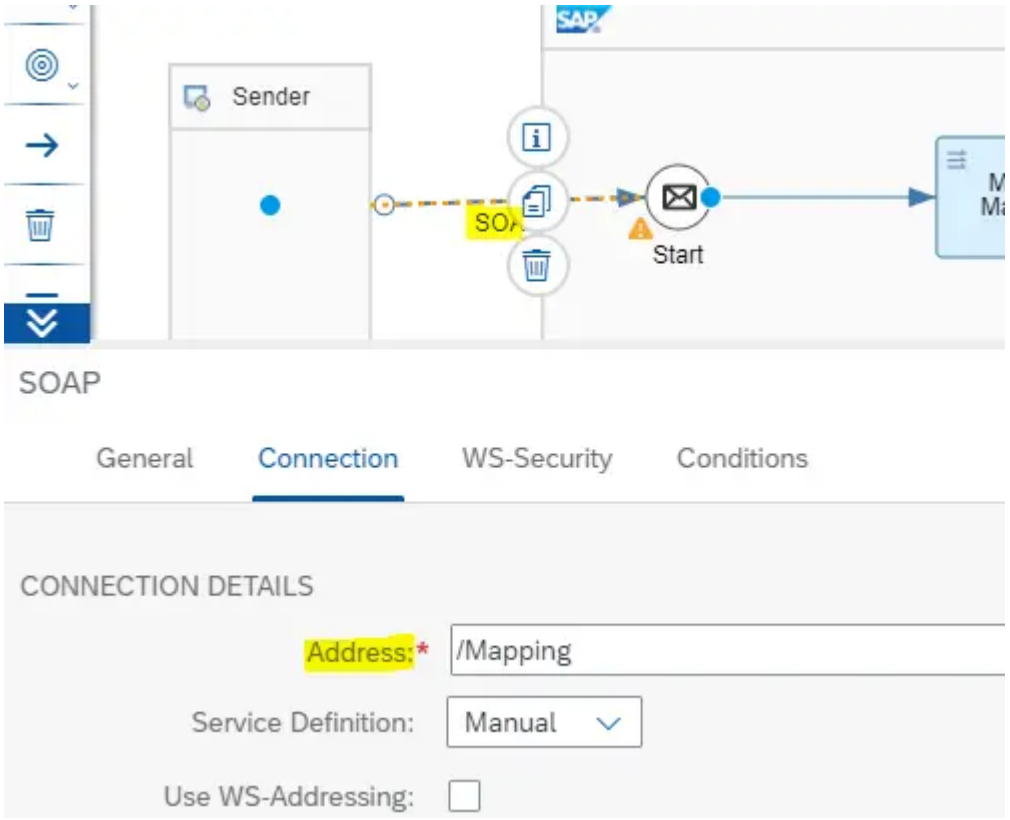
Choose **Sender Adapter** as **SOAP**.

Back To Top

Menu



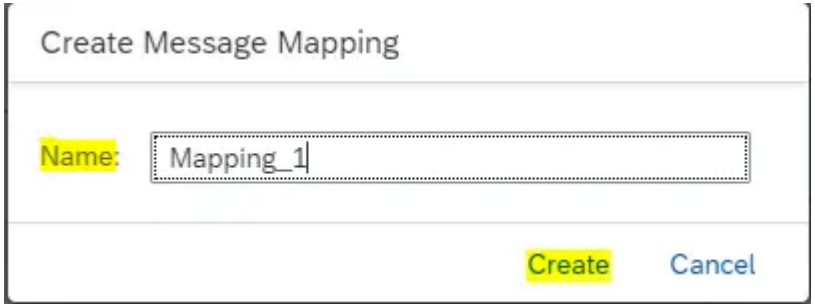
Step 11 : Select **SOAP Adapter**. Under ***Connection*** tab, Give a relative ***address*** to define the Endpoint. (/Mapping)



Step 12 : Select **Message Mapping** and Click on **Create** button highlighted below.

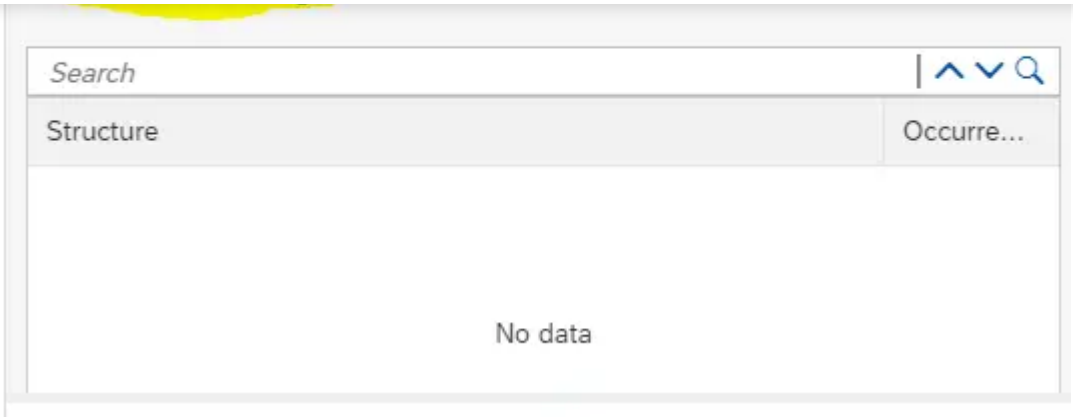
Back To Top

Step 13 : Give any Valid ***name*** for your Mapping. Click on **Create**.



Step 14 : Click on **Add Source Message**.

Back To Top



Step 15 : Click on ***Upload from File System***. Browse and select the **Source WSDL.**

Copy the below sample WSDL for Source message and save to upload.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="SendOrder_Async"
    targetNamespace="http://cpi.sap.com/demo" xmlns:p1="http://cpi.sap.com/demo"
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:documentation />
    <wsp:UsingPolicy wsdl:required="true" />
    <wsp:Policy wsu:Id="OP_SendOrder_Async" />
    <wsdl:types>
        <xsd:schema targetNamespace="http://cpi.sap.com/demo"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://cpi.sap.com/demo">
            <xsd:element name="Order_MT" type="Order_DT" />
            <xsd:complexType name="Order_DT">
                <xsd:sequence>
```

Back To Top

```
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:schema>
        </wsdl:types>
        <wsdl:message name="Order_MT">
            <wsdl:documentation />
            <wsdl:part name="Order_MT" element="p1:Order_MT" />
        </wsdl:message>
        <wsdl:portType name="SendOrder_Async">
            <wsdl:documentation />
            <wsdl:operation name="SendOrder_Async">
                <wsdl:documentation />
                <wsp:Policy>
                    <wsp:PolicyReference URI="#OP_SendOrder_Async" />
                </wsp:Policy>
                <wsdl:input message="p1:Order_MT" />
            </wsdl:operation>
        </wsdl:portType>
        <wsdl:binding name="SendOrder_AsyncBinding" type="p1:SendOrder_Async">
            <soap:binding style="document"
                                                    transport="http://schemas.xmlsoap.org/soap/http"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
            <wsdl:operation name="SendOrder_Async">
                <soap:operation soapAction="http://sap.com/xi/WebService/soap1.1"
                    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
                <wsdl:input>
                    <soap:body use="literal" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
                </wsdl:input>
            </wsdl:operation>
```

Back To Top

Click on Upload from the file system.

Step 16 : **Repeat** the **Step 14 and Step 15** for **Target message** and choose target WSDL.

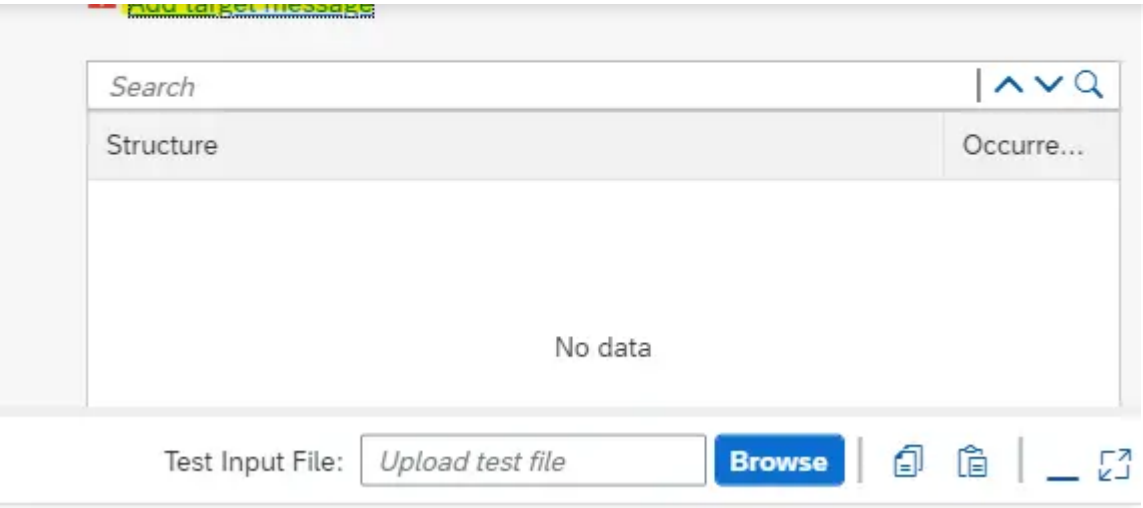
Copy the below sample WSDL for Target message and save to upload.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="SendOrder_Async"
    targetNamespace="http://cpi.sap.com/demo" xmlns:p2="http://cpi.sap.com/demo"
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:documentation />
    <wsp:UsingPolicy wsdl:required="true" />
    <wsp:Policy wsu:Id="OP_SendOrder_Async" />
    <wsdl:types>
        <xsd:schema targetNamespace="http://cpi.sap.com/demo"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://cpi.sap.com/demo">
            <xsd:element name="Order_MT" type="Order_DT" />
            <xsd:complexType name="Order_DT">
                <xsd:sequence>
                    <xsd:element name="orderNumber" type="xsd:string" />
                    <xsd:element name="Environment" type="xsd:string" />
                </xsd:sequence>
            </xsd:complexType>
        </xsd:schema>
    </wsdl:types>
```

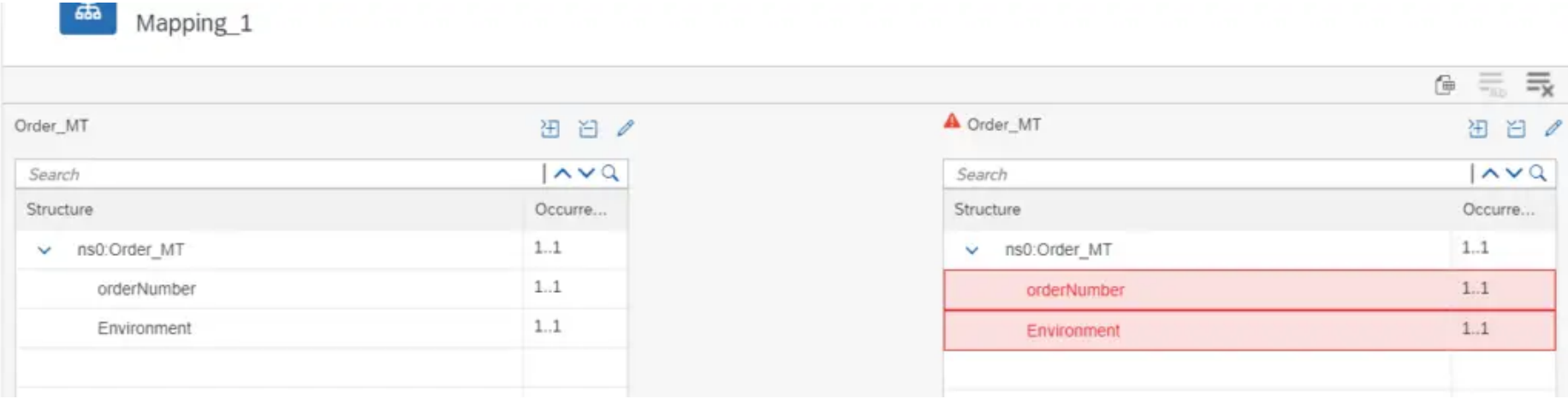
Back To Top

```
        <wsdl:part name="Order_MT" element="p2:Order_MT" />
    </wsdl:message>
    <wsdl:portType name="SendOrder_Async">
        <wsdl:documentation />
        <wsdl:operation name="SendOrder_Async">
            <wsdl:documentation />
            <wsp:Policy>
                <wsp:PolicyReference URI="#OP_SendOrder_Async" />
            </wsp:Policy>
            <wsdl:input message="p2:Order_MT" />
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="SendOrder_AsyncBinding" type="p2:SendOrder_Async">
        <soap:binding style="document"
                                                    transport="http://schemas.xmlsoap.org/soap/http"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
        <wsdl:operation name="SendOrder_Async">
            <soap:operation soapAction="http://sap.com/xi/WebService/soap1.1"
                xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
            <wsdl:input>
                <soap:body use="literal" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
            </wsdl:input>
        </wsdl:operation>
    </wsdl:binding>
</wsdl:definitions>
```

Back To Top

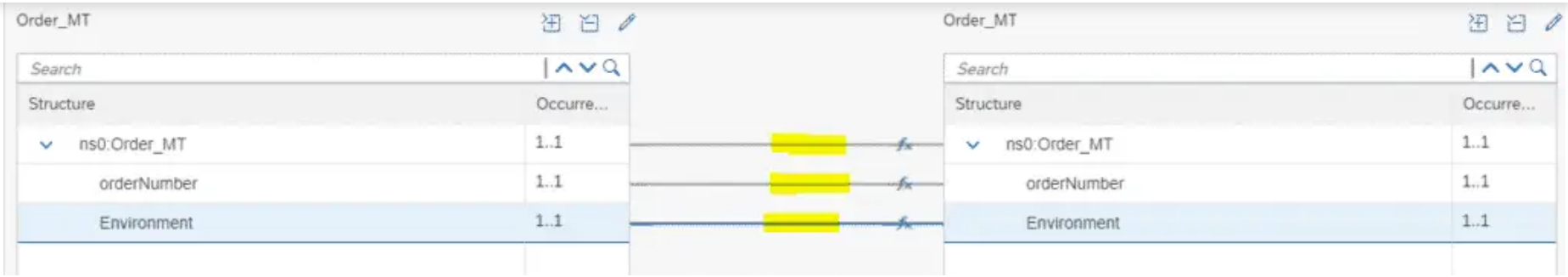


Step 17 : It will look as below.



Step 18 : Click on Source Element and drag to the target element to map respective fields. Now it will look as below.

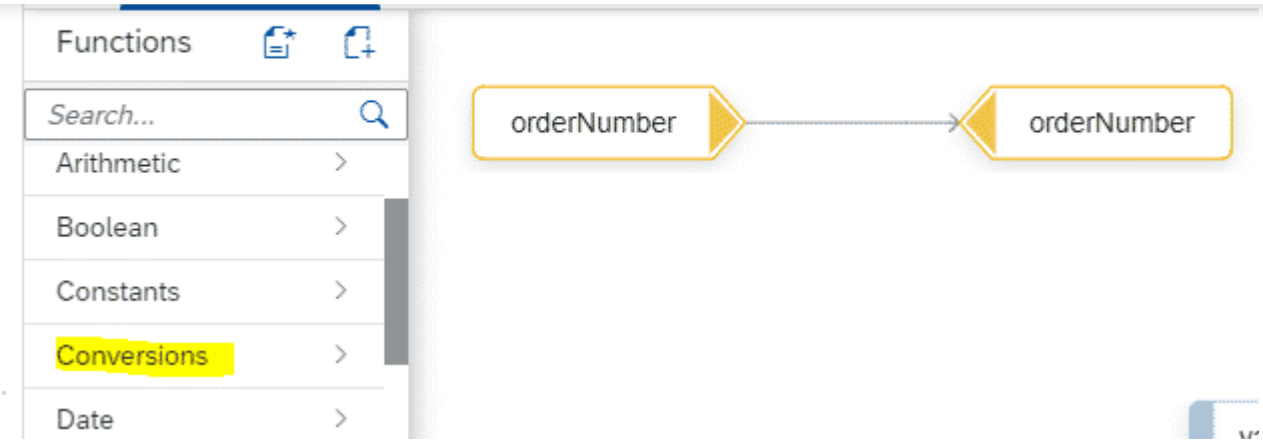
Back To Top



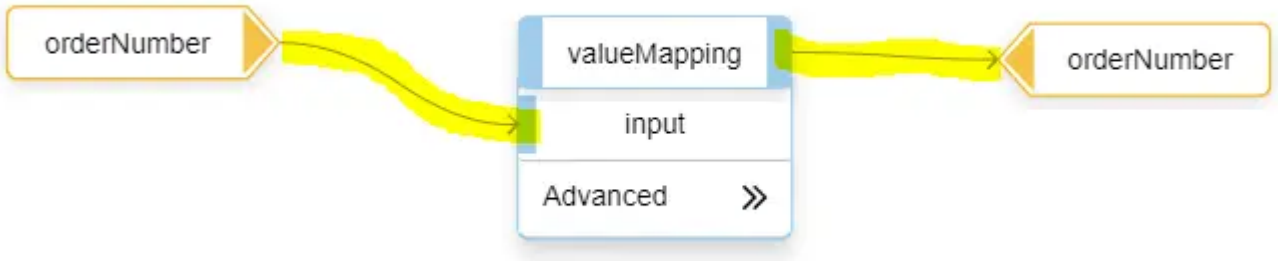
Step 19 : Click on **fx** Symbol to edit the mapping for that element.

Step 20 : From the left window, choose **Conversions –> Value Mapping**

Back To Top



Step 21 : Place the Value Mapping block in center and **Connect Source to input** , **valueMapping to Target** as shown in the below picture.



Step 22 : Click on **Advanced** and fill the required details same as the **deployed Value Mapping** in Step 7.

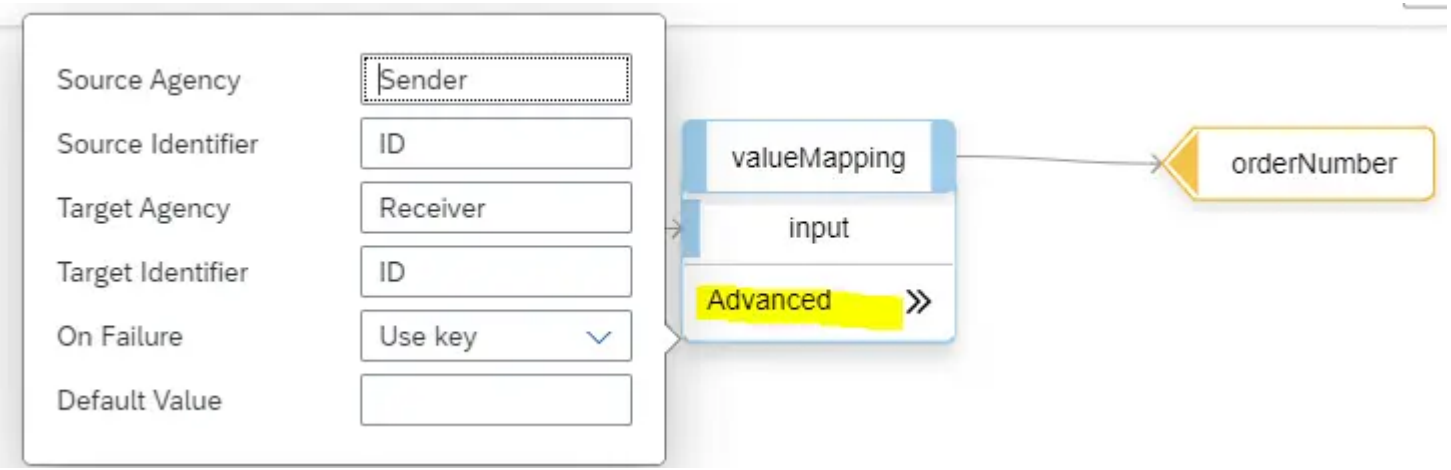
On Failure :

On Failure	Expected Outcomes when the incoming Message has any other value apart from the once maintained in the Value Mapping Used (in our case any value other than A, B , C)

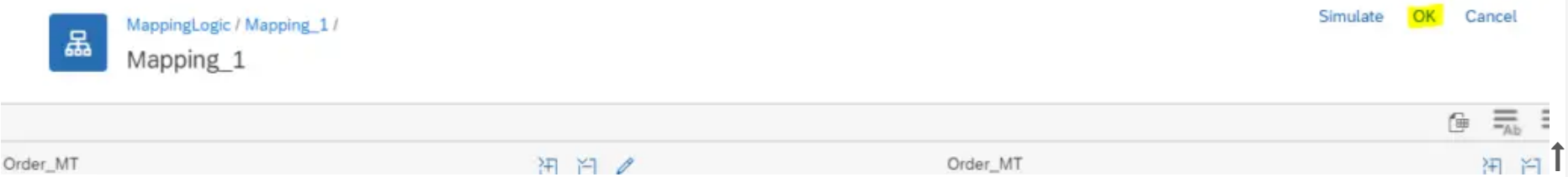
Back To Top

Use Default Value	It will pass the Default Value that is specified in the Default Value field.
Throw Exception	This will throw an exception.

I have used Use Key.



Step 23 : Click on **OK** at top right corner. It will redirect you to the Main Integration Flow.

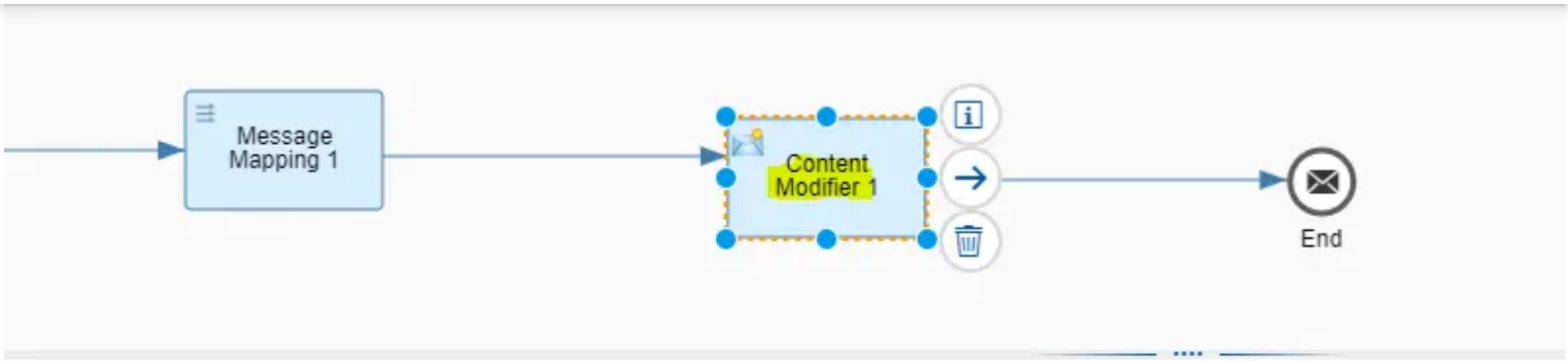


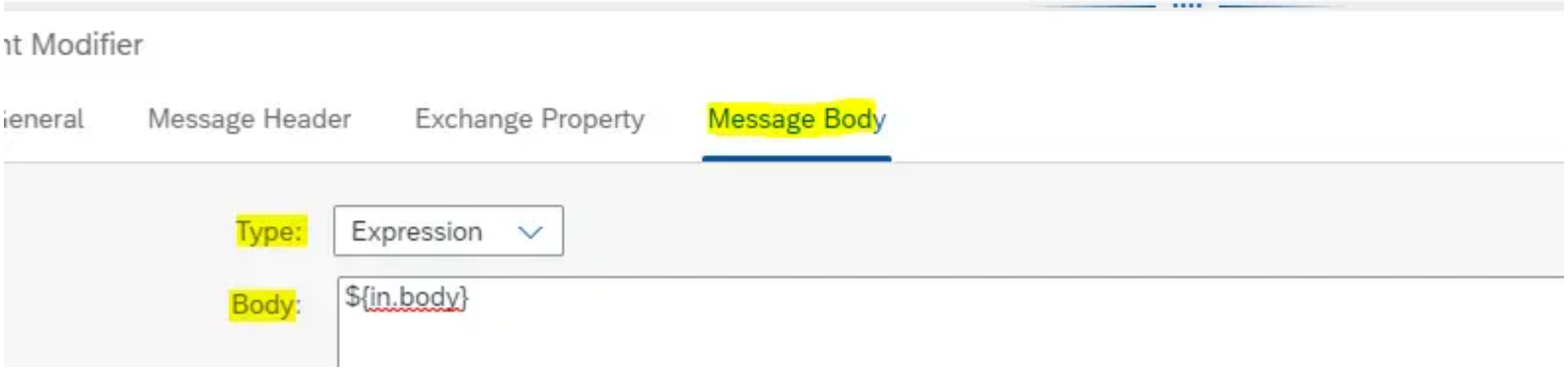
Step 24 : You can **edit** your mapping any time from **Processing** tab of Message Mapping.

Back To Top

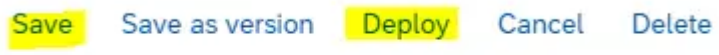
Step 25 : Select the **Content Modifier**. Under **Message Body** tab, choose **Type=Expression** and **Body : ${in.body}** to capture the incoming payload.

Back To Top



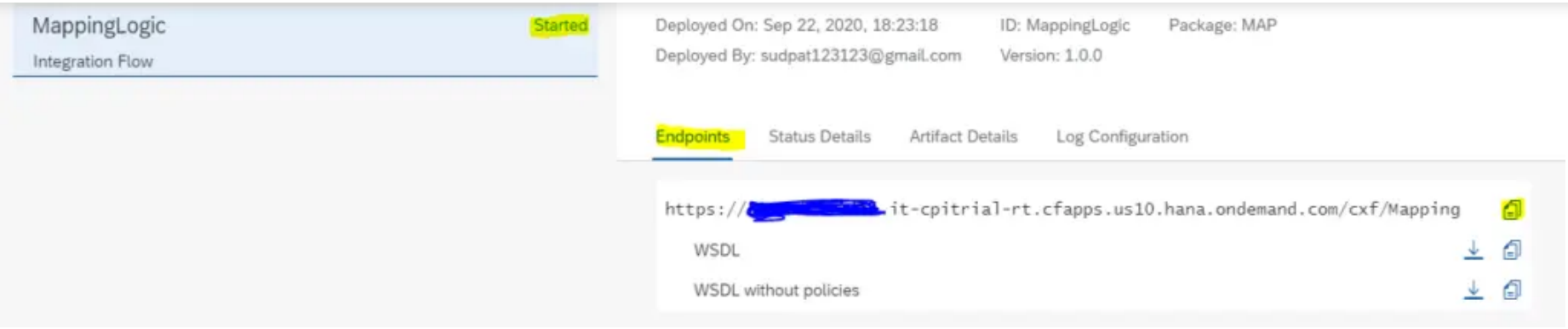


Step 26 : **Save** and **Deploy.**



Step 27 : In **overview** window, make sure that your flow is in **Started** state. Click on **Copy button** highlighted under ***Endpoints*** tab to Copy the Endpoint.

Back To Top



TESTING THE ARTIFACTS

Step 28 : Open **POSTMAN**. Create a new request with type **POST**.

Paste the **Endpoint** copied in previous step in **URL** tab.

Type : Basic Auth

Username : Client ID

Password : Client Secret

Client Id and Client Secret you will get when you create Process Integration Runtime Instance during tenant setup. (Refer **Step 34** of this post)

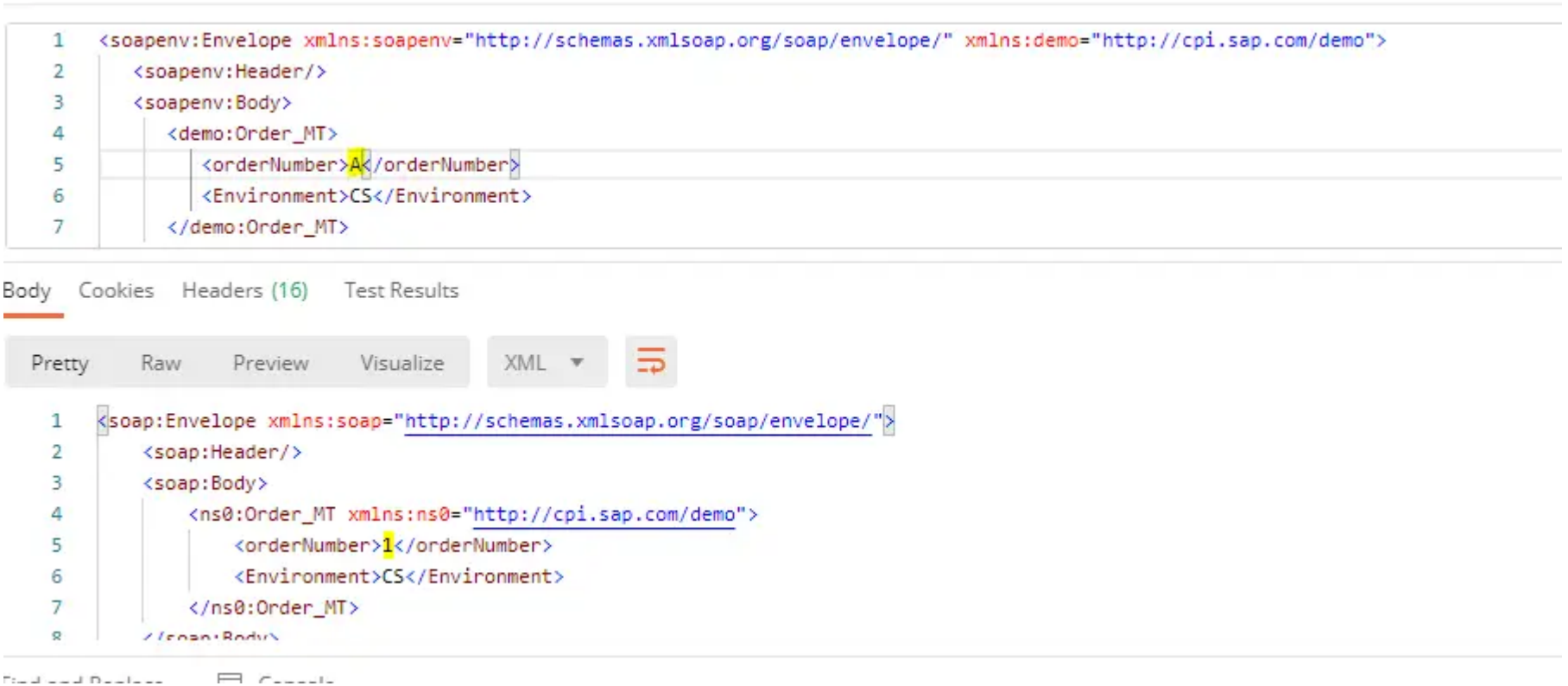
Back To Top

Step 29 : Under **Body** tab, Choose **Raw** radio button and **XML** message Type. **Type** the below input message. Click on **Send**.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:demo="http://cpi.sap.com/demo">
  <soapenv:Header/>
  <soapenv:Body>
    <demo:Order_MT>
      <orderNumber>A</orderNumber>
      <Environment>CS</Environment>
    </demo:Order_MT>
  </soapenv:Body>
</soapenv:Envelope>
```

Back To Top

Step 30 : You will get back the result as below. Note that the ***OrderNumber is changed from A to 1 in output***. This is due to the use of Value Mapping.



Step 31 : Change **orderNumber** to **B** and **C** and click on **Send**. You should get the response as **2** and **3** respectively.

Back To Top

Menu

SAP Training Class

Join Us Now

Awarded No.1 SAP Education Partner. 15+ Years of Experience. Apply Now.

sapeducation.atos.net

OPEN

none form-data x-www-form-urlencoded raw binary GraphQL XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:demo
   <soapenv:Header/>
   <soapenv:Body>
      <demo:Order_MT>
         <orderNumber>45</orderNumber>
         <Environment>CS</Environment>
      </demo:Order_MT>
```

Body Cookies Headers (16) Test Results

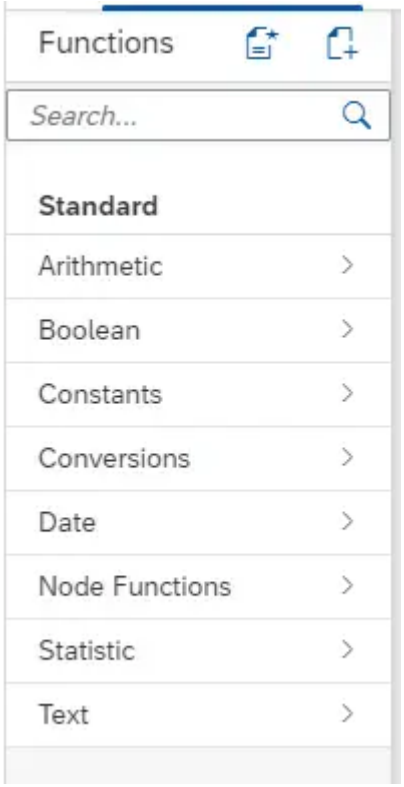
Pretty Raw Preview Visualize XML

```
   <soap:Body>
      <ns0:Order_MT xmlns:ns0="http://cpi.sap.com/demo">
         <orderNumber>45</orderNumber>
         <Environment>CS</Environment>
      </ns0:Order_MT>
   </soap:Body>
</soap:Envelope>
```

SELF- TRY :

Back To Top

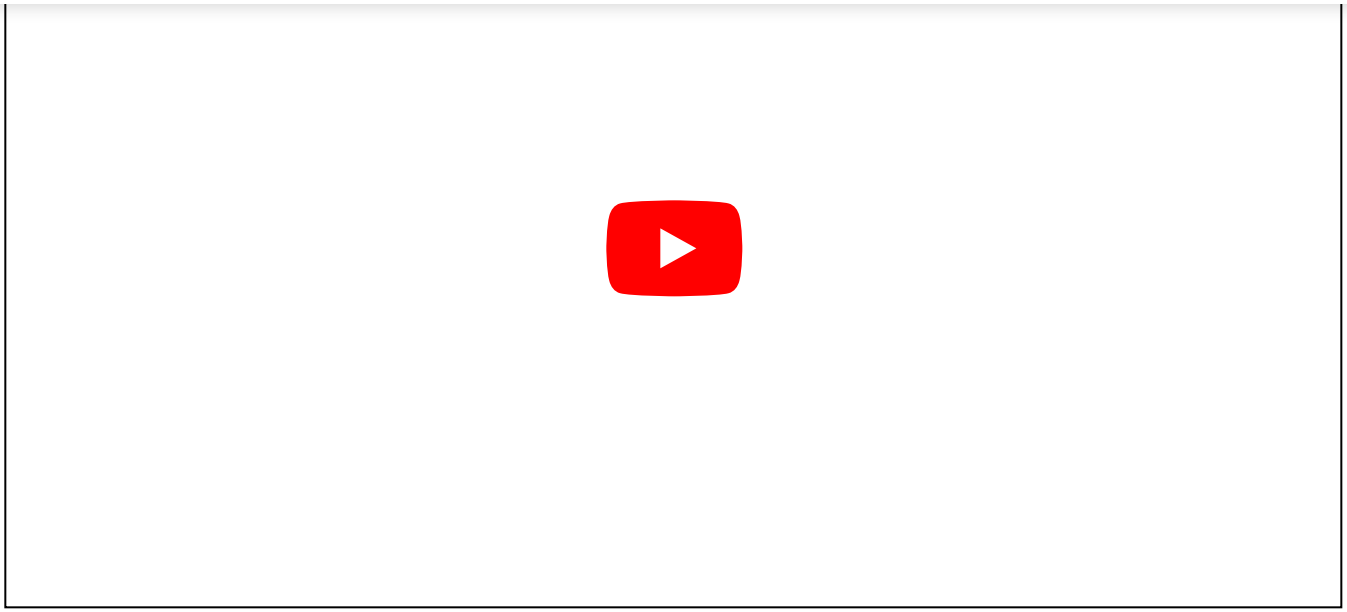
2. Try adding different functions available in Mapping window to transform source to target and test the same.



SAP CPI Message Mapping Tutorial

The below is a detailed example with SAP CPI message mapping using if conditions, and how to do namespace mapping. If you are good to go with Message mapping in CPI let's look into a bit advanced version which is importing value mapping to SAP CPI.

Back To Top



FINAL VERDICT :

In conclusion, I hope you enjoyed reading this article on "**Message Mapping and Value Mapping in SAP CPI**", If yes, then don't forget to spread the word about it. Do send the feedback and to know more about it. Also, have a sneak peek at our SAP CPI Tutorials. Signing off Sudarshan@recodehive.com

Adaptation	Growth Hacking	Developer Bundle	Hope and Focus	My Day at Google Campus	Time Machine	Manifestation of life after death

What is Multi Mapping in SAP CPI?

It's a way of using multiple message in sender or reciever system for your message mapping.

How Message Mapping work?

It uses Internal Queues, basically these are XML structure of the source message and nodes of these function is categorised into context. This can be use to trace the programs etc.

Back To Top

0 comments

Sort by Newest

Add a comment...

Facebook Comments Plugin

32 66 4 1.5K

70
Shares

Tagged CPI, Learning, Mapping, SAP, SAP CPI

RELATED POSTS

Back To Top

Header based Routing & Local Integration Process

Importing Value Mapping in SAP CPI

Back To Top

Back To Top

Back To Top

Back To Top

Back To Top

Back To Top

Back To Top

Back To Top

Back To Top

Back To Top

Back To Top

Back To Top

Back To Top

Menu

Back To Top

Menu

Back To Top

Back To Top

Back To Top

Back To Top

Menu

Back To Top

Recode Hive © 2022 | All right reserved