Interface Functional Design

Sales Quote Creation Functional Design

CallidusCloud CPQ C4C Integration Project

Version 0.7

Table of Contents

Document Information **3**

Overview and Scope **4**

Functional Overview 4

Detailed Requirements 4

Gap in Standard SAP Filled by This Item 4

Solution Alternatives 4

Proposed Solution 4

Business Benefit 4

Scope 4

Functionality 4

Region or Localization Considerations 4

Dependencies/Constraints 5

Assumptions 5

Detailed Functional Requirements **5**

Functional Specification 5

Process Flow Diagram 9

Initiating Process/Transaction/Program 9

Security Requirements 9

System Details and Direction 9

Proposed Transfer Method / Program 9

Field Mapping from Source to Destination 10

Post-Interface Activities 10

Custom Tables/Fields Specifications 10

Initial Data Setup, Conversion Requirements 10

Configuration Requirements 10

Language Considerations 10

Reporting 10

Error Handling 10

Security and Authorization 10

Reconciliation Procedures and Audit Requirements 10

Processing and Operational Considerations 11

Technical Specifications 12

Creation of Communication System 12

Creation of Communication Scenario 12

Creation of Communication Arrangement 12

Overview of the Process Steps 13

Create an Integration project in SCP IS (HCI) 13

Develop the integration flow diagram 15

Import WSDLs and schema files 22

Create message mapping 22

Create Scripts 24

Configure Channels 24

Create Value Mapping 26

UpdateOpportunityAndSalesQuote Main IFlow 29

Deployment of Artifact 31

1 Document Information

FD Document Name	Sales Quote Creation
Interface Name	Sales Quote Creation
Sub-Process	
Complexity	Very High
Source System	CallidusCloud CPQ
Destination System	SAP Hybris C4C
Target Environment	

2 Overview and Scope

2.1 *Functional Overview*

This functional design details the process to create and/or update sales quotes in SAP Hybris Cloud for Customer (SAP Hybris C4C) from a quotation created in CallidusCloud CPQ. It defines the interface that is designed to update pricing and product related information in SAP Hybris C4C sales quote from the associated Callidus CPQ quotation.

2.1.1 Detailed Requirements

Requirement ID	Business Process/Stream	Short Description	Description	Requirement Type
	Sales Quote	Sales Quote Creation	Sales Quote Creation From CPQ	Integration

2.1.2 Gap in Standard SAP Filled by This Item

N/A

2.1.3 Solution Alternatives

N/A

2.1.4 Proposed Solution

Standard interface to be used to replicate pricing and product details from CallidusCloud CPQ quotation to SAP C4C sales quote.

2.2 *Business Benefit*

Business users will be able to synchronize product and pricing information between Callidus CPQ quote to SAP C4C sales quote in real time.

2.3 *Scope*

2.3.1 Functionality

- This interface is designed to replicate calculated price and product details from CallidusCloud CPQ quote to SAP C4C sales quote.
- Attachments are set to come to C4C sales quote from Callidus CPQ. Details of this functionality are out of the scope for this document and are covered in a corresponding functional design document.

2.3.2 Region or Localization Considerations

NA

2.4 *Dependencies/Constraints*

- Business users should have access to SAP Hybris C4C, SAP Cloud IDP and CallidusCloud CPQ application.

2.5 *Assumptions*

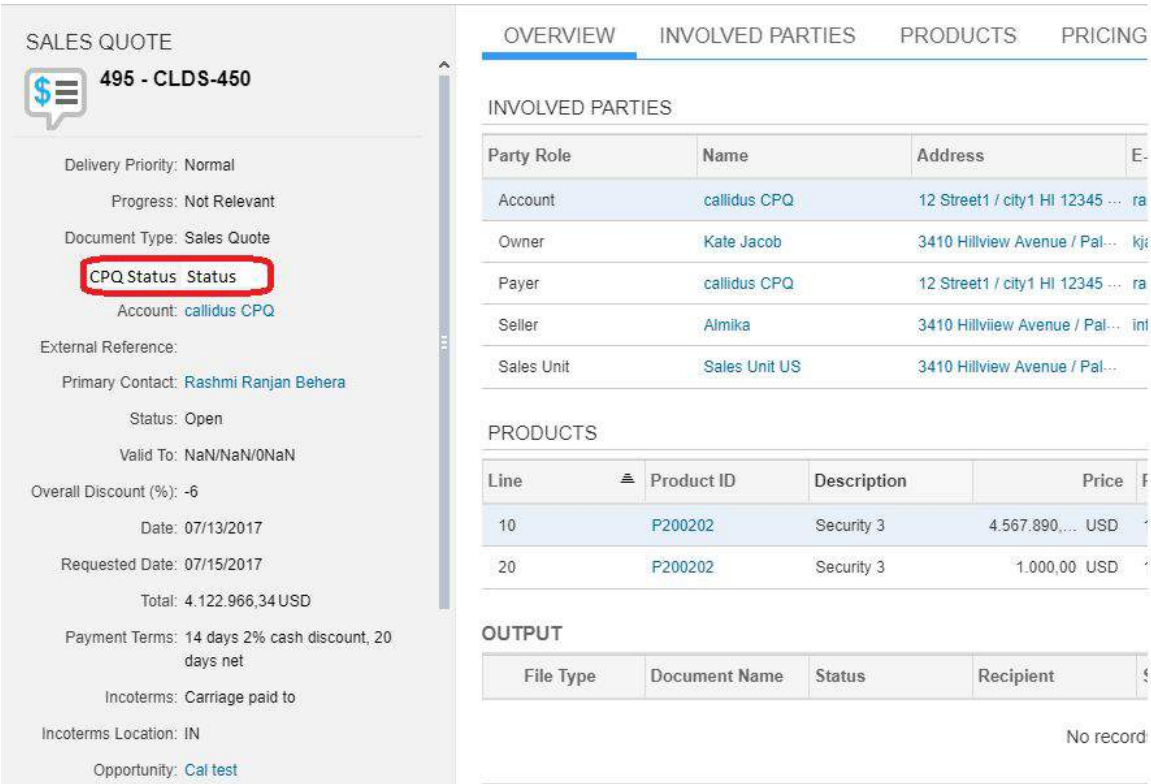
- Product master is in sync between SAP Hybris C4C and CallidusCloud CPQ system.

3 Detailed Functional Requirements

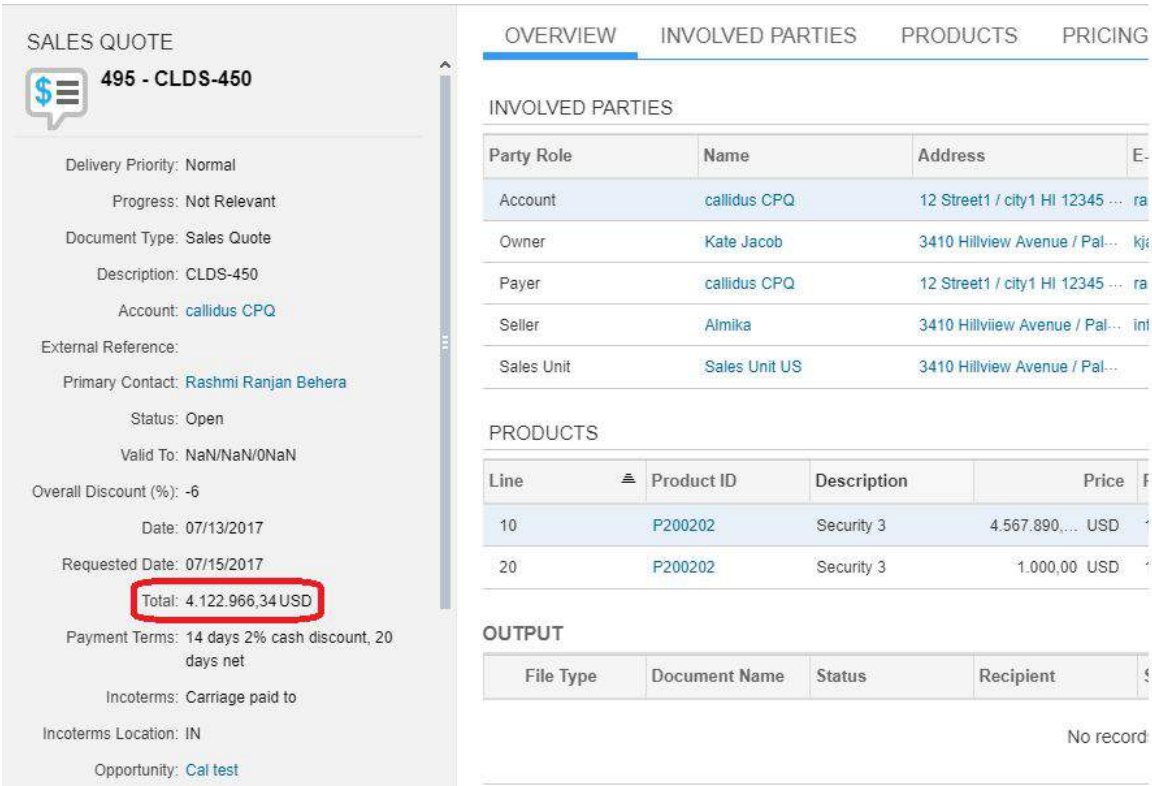
3.1 *Functional Specification*

Sales quote creation from CPQ

- Sales quote is created in CPQ.
- Sales quote is created at the discretion of a user.
- The request for sales quote creation is initiated in CPQ and, hence, it is inbound to C4C.
- Field default values to be assigned at the middleware (SCP IS).
    - Document type for this implementation is defaulted at the middleware.
- New custom field is created using KUT (Key User Tool) at the header level of the C4C sales quote to accommodate the value of CPQ quotation status.



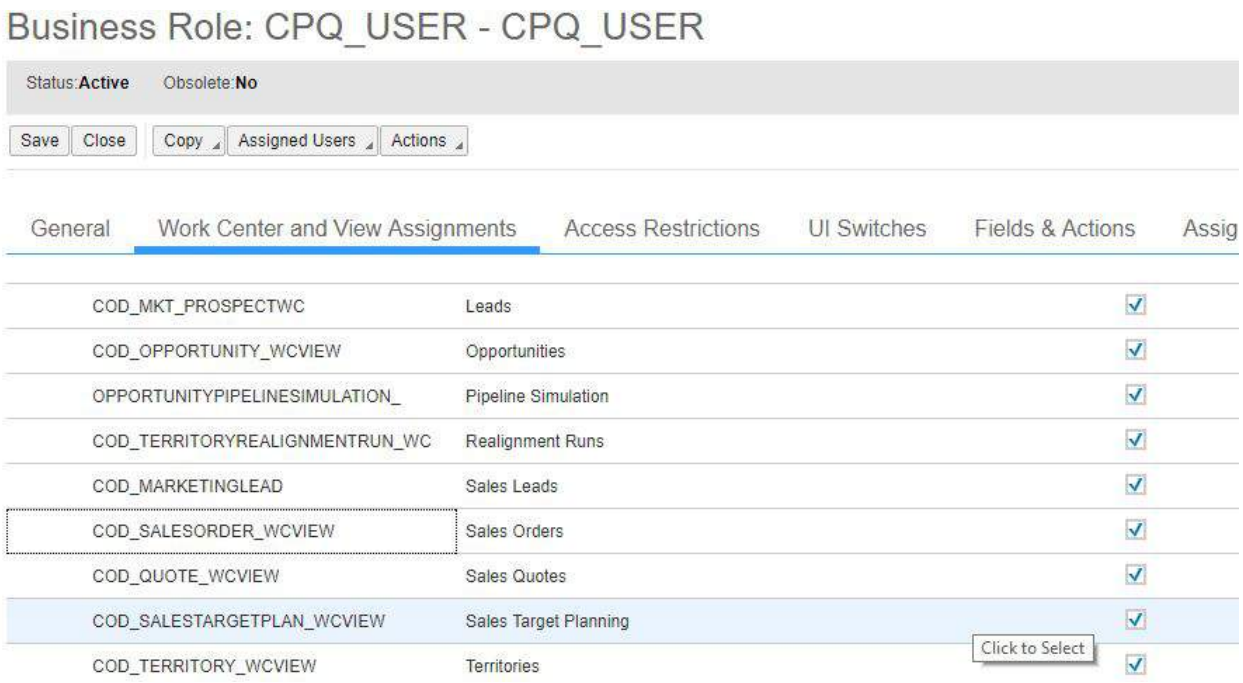
- Value Mapping for price component to be performed at the middleware.
- CPQ sends the price in a predefined structure with the list price and discounts for each item as well as the overall discount at the header level.
    - For the quote creation, quote header with items is replicated first then the price components are replicated to C4C from CPQ.
    - For quote update, Items are cleared first, and then the item list, which comes in payload, is added along with price components.
- The field for Total value of the quotation at header level is non-editable and is a calculated result of summation of all line items (products), plus the overall discount.



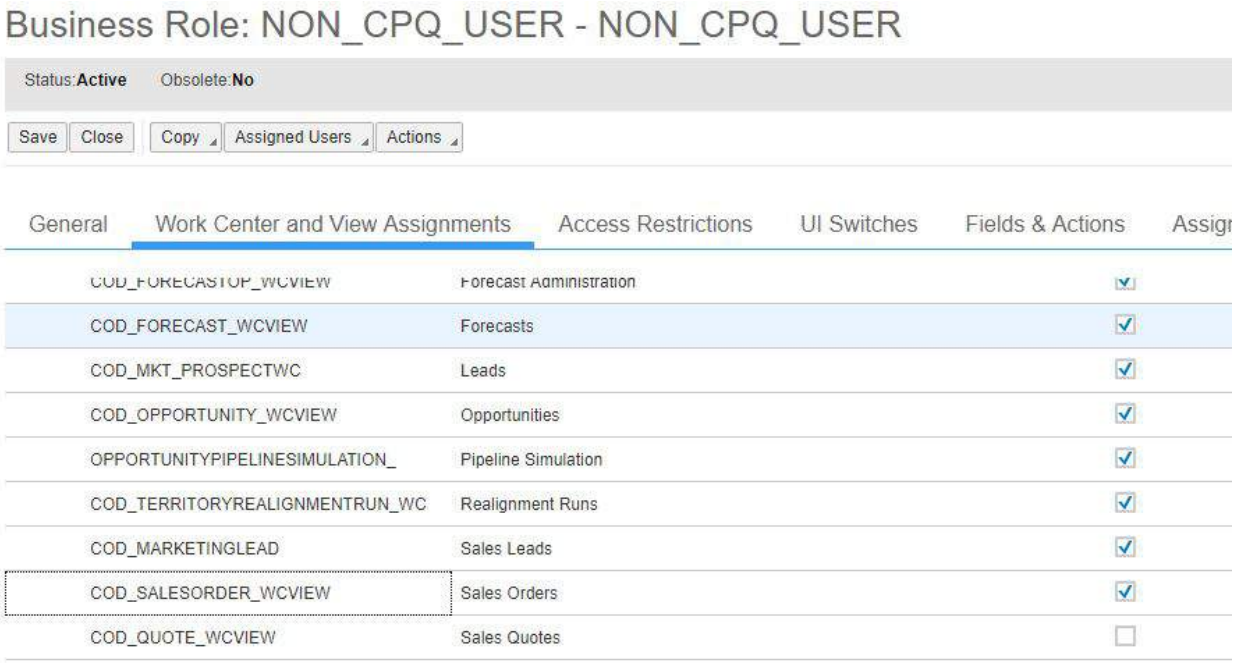
Business Role Creation

Two Business Roles will be created.

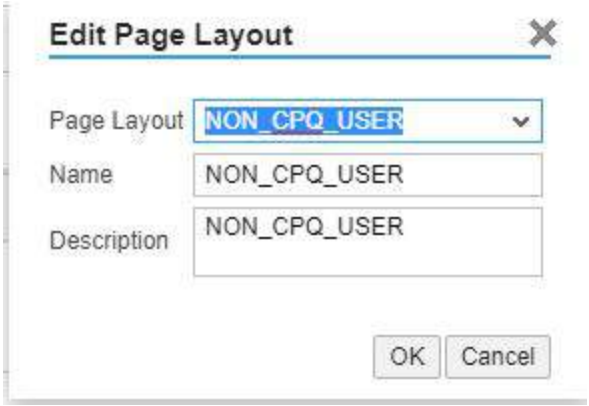
- CPQ_USER – This role includes an access to C4C sales quote view.

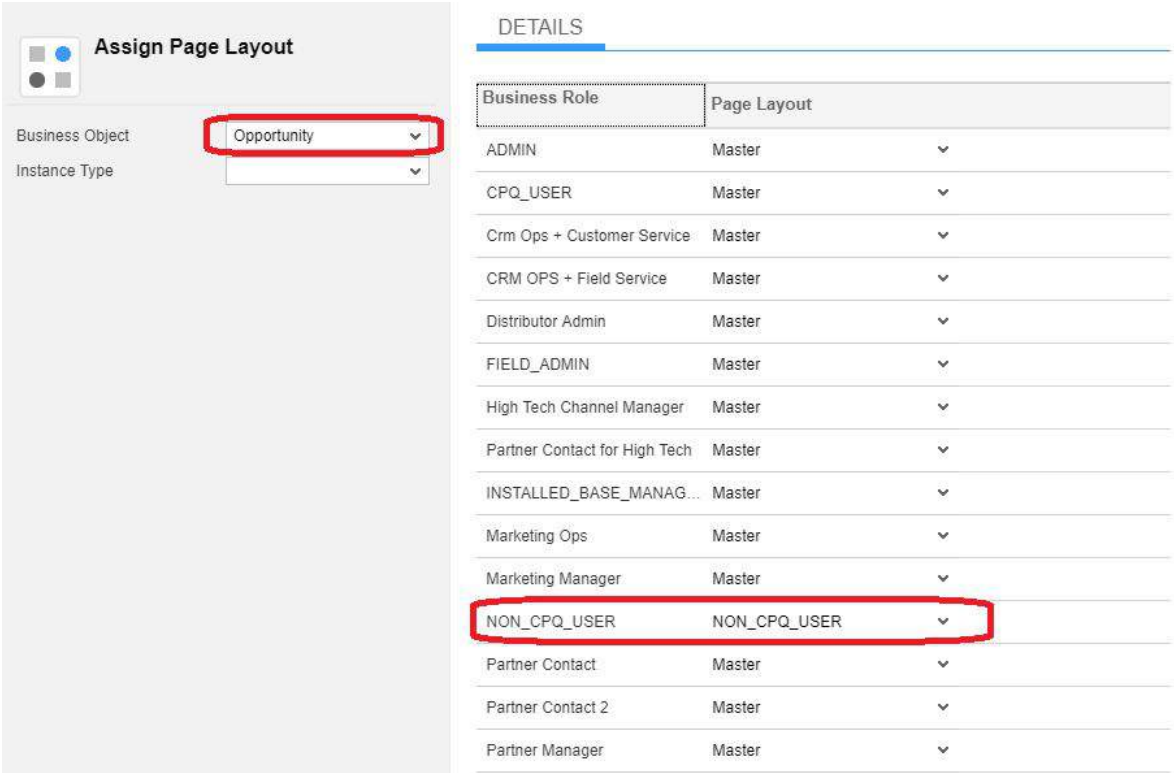


- NON_CPQ_USER – This role excludes sales quote view access.

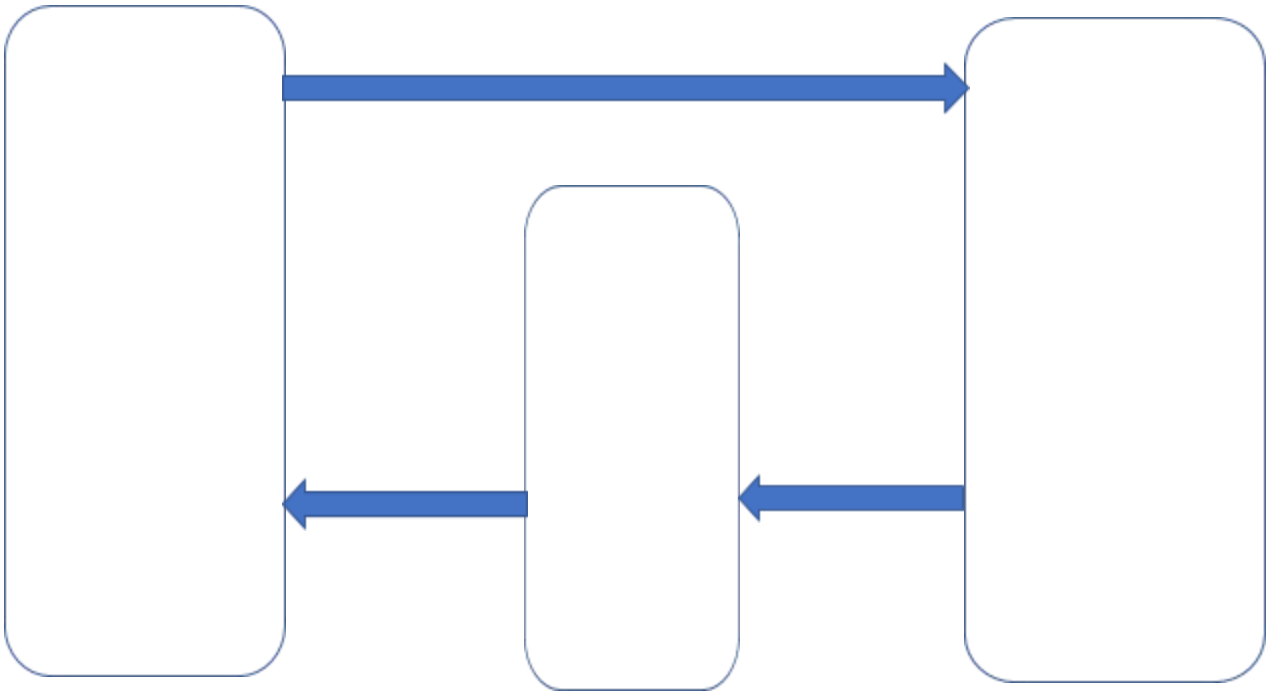


- This business role – when assigned users – will prevent them from accessing the tab (facet) for the sales quotes view. However, the users can still read and view sales quotes from inside opportunities by selecting a menu option “Sales Documents”. To eliminate such a possibility, a new page layout should be created for assigning a business role. In this page layout, the option to select sales quote will be removed in Sales Documents.





3.2 *Process Flow Diagram*



3.3 *Initiating Process/Transaction/Program*

N/A

3.4 *Security Requirements*

N/A

3.5 *System Details and Direction*

Interface Type: A2X (Application to X User, Synchronous)
Direction: Inbound to SAP C4C
Source System: Callidus Cloud CPQ system
Destination System: SAP Hybris C4C
Middleware: SAP Cloud Platform Integration Services

3.6 *Proposed Transfer Method / Program*

Transfer Method (SAP): Web Service (SOAP)

3.7 *Field Mapping from Source to Destination*

3.8 *Post-Interface Activities*

N/A

3.9 *Custom Tables/Fields Specifications*

N/A

3.10 *Initial Data Setup, Conversion Requirements*

List of product categories and products must be in sync between SAP Hybris C4C and CallidusCloud CPQ.

3.11 *Configuration Requirements*

N/A

3.12 *Language Considerations*

N/A

3.13 *Reporting*

N/A

3.14 *Error Handling*

Potential Errors:

Notification Procedures:

Contingency Procedures:

3.15 *Security and Authorization*

No special security required.

3.16 *Reconciliation Procedures and Audit Requirements*

3.16.1 **Approach**

3.16.2 **Metrics**

3.17 *Processing and Operational Considerations*

3.17.1 Processing Type

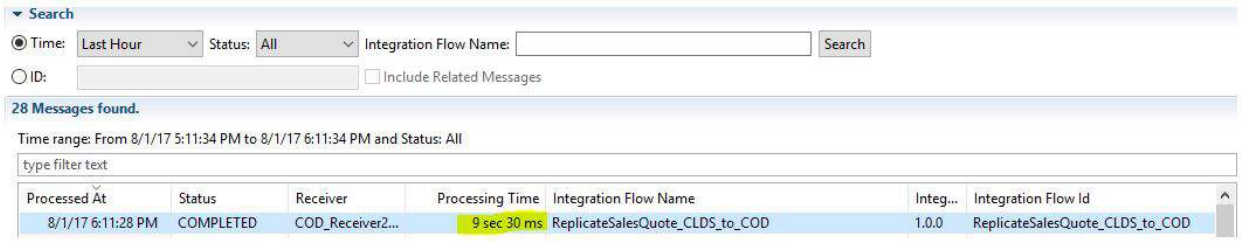
3.17.2 Transaction Volume

3.17.3 Frequency and Timing

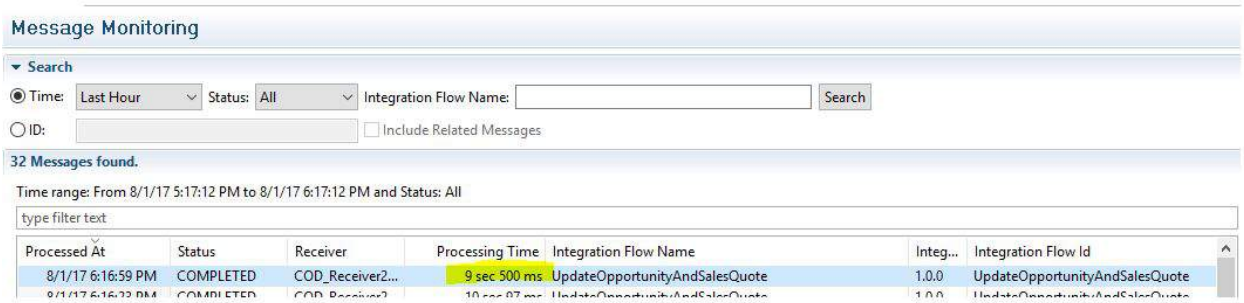
Real-time update of a C4C sales quote from the associated Callidus CPQ quote

3.17.4 Performance

CreateUpdateSalesQuote – 9 s 30 ms



UpdateOpportunityAndSalesQuote – 9 s 500 ms



3.17.5 Batch Requirements

N/A

3.17.6 Data Maintenance Requirements

N/A

3.17.7 Re-use Details

N/A

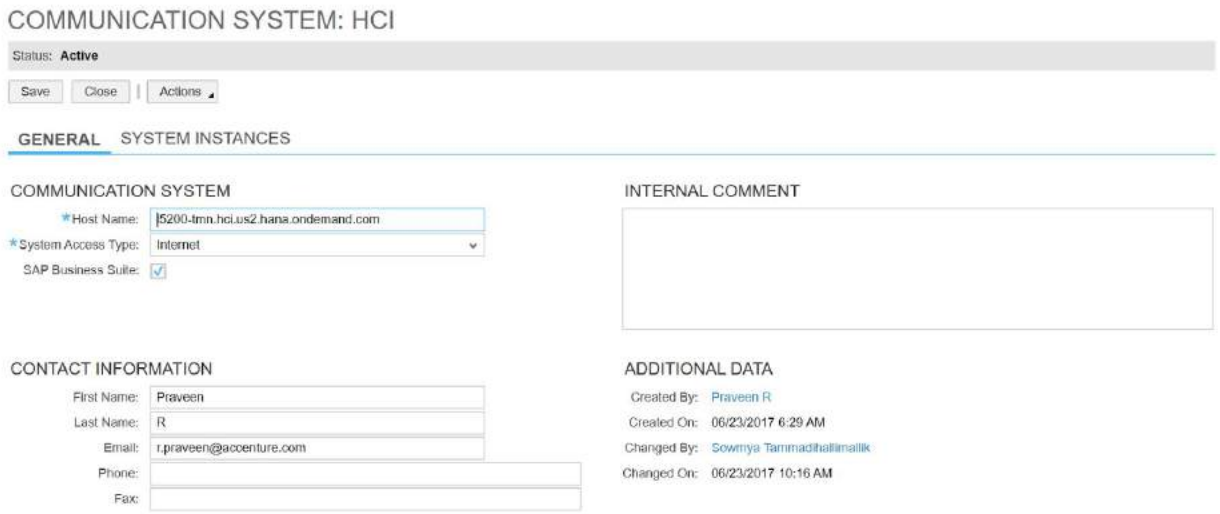
3.17.8 Multi-company, Multi-jurisdiction Considerations

None

4 Technical Specifications

4.1 *Creation of Communication System*

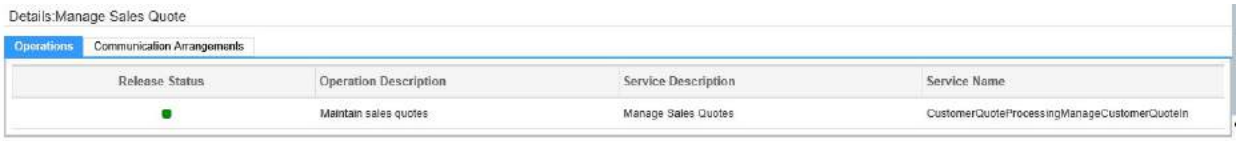
For exchange of data between SAP Hybris C4C and CallidusCloud CPQ, a communication system is created in C4C with the host name provided as the URL of SCP IS (HCI) system.



4.2 *Creation of Communication Scenario*

Inbound and Outbound services for a business process are defined in Communication Scenario. SAP Hybris C4C does not have a standard scenario for utilizing a sales quote A2X service(s).

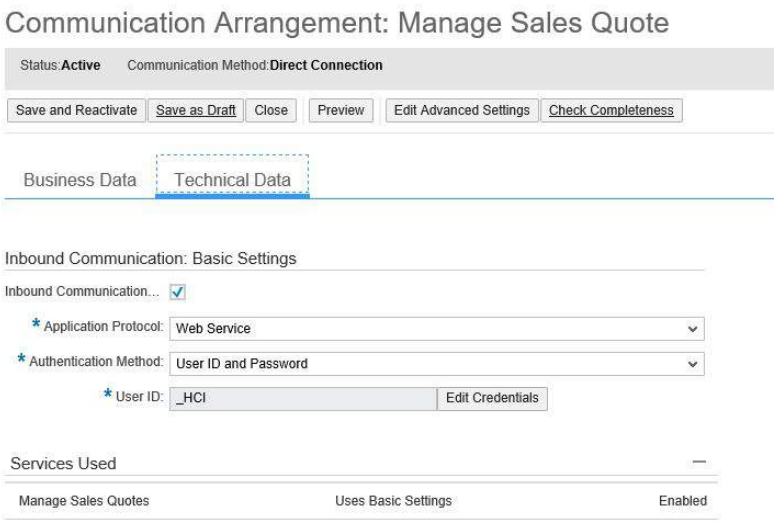
Task: Create a new communication scenario using “Manage Sales Quote” service to be used in the corresponding communication arrangement.



4.3 *Creation of Communication Arrangement*

Task: Create communication arrangement to link communication scenario “Manage Sales Quote” and SCP IS (HCI) created in previous steps.

Set an authentication method to the basic, i.e. “User ID and Password” and maintain the password.



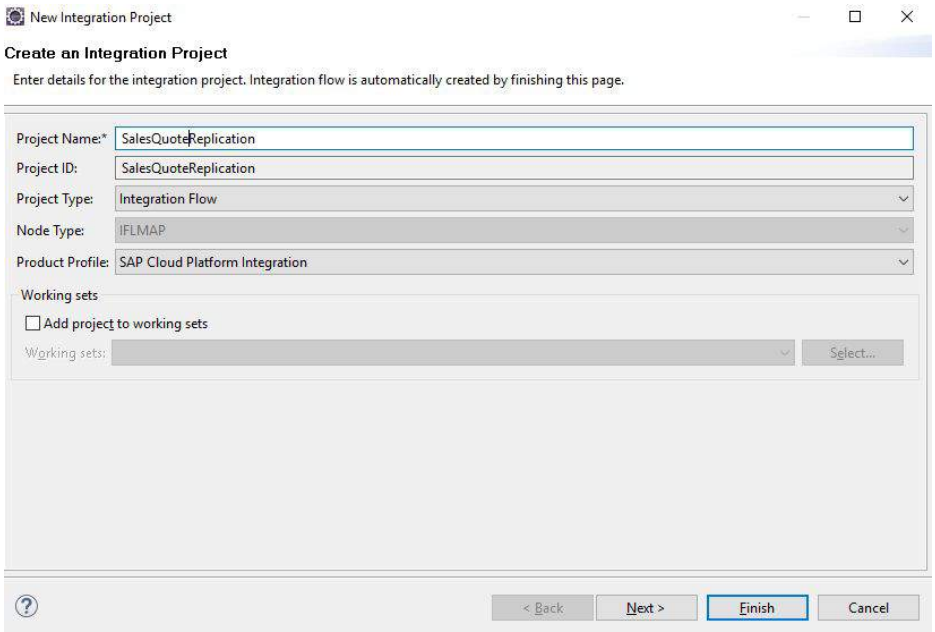
4.4 *Overview of the Process Steps*

Process Step No.	Name	Description	Classification
PS1	Create Integration Project in SCP IS (HCI)	Create a new integration project in SCP IS (HCI) in Eclipse	New
PS2	Develop Integration Flow Diagram	Model iFlow in accordance to the requirements	New
PS3	Import WSDLs and Schema Files	Import all files used for the message mapping	New
PS4	Create Message Mapping	Map source and target fields for data flow	New
PS5	Create Scripts	Groovy scripting is used at some places to fully meet the requirements	New
PS6	Configure Channels	Configure sender and receiver channels	New
PS7	Create Value Mapping	Value mapping is required to convert values into C4C Internal Code and vice versa	New
PS8	UpdateOpportunityAndSalesQuote Main IFlow	Multi Casting process to Update Opportunity and Quote at the same time	New
PS9	Deploy Artifact	Deploy the developed artifact to iFlow	New

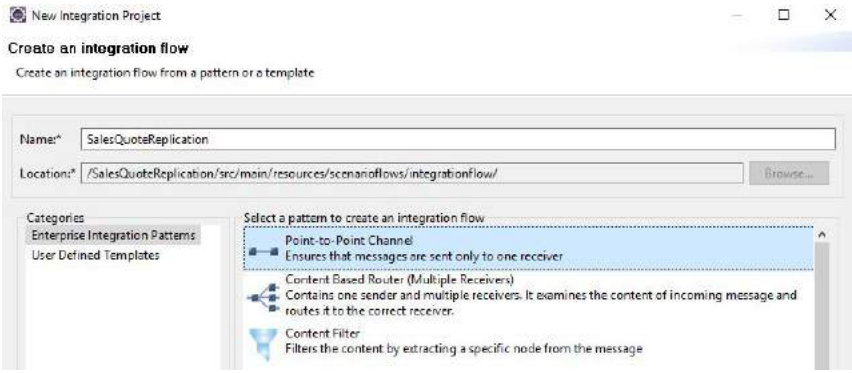
4.4.1 **Create an Integration project in SCP IS (HCI)**

Classification	☒ New ☐ Change of Existing Step ☐ Reuse of Existing Step
Objective	To create a new integration project in HCI for Sales Quote Replication
Involved Users	
Trigger	N/A
Precondition	SCP IS (HCI) developer is provided with the authorization to create artifact
Post-condition	N/A
End User Interaction with the System	End user is not involved directly

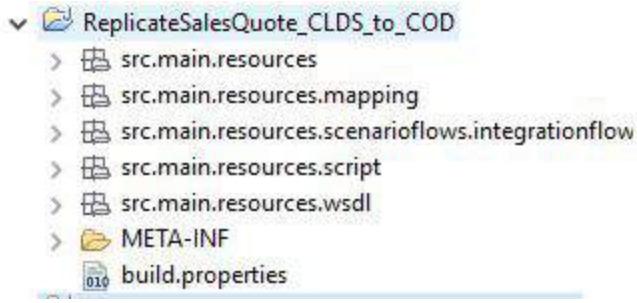
1. Create an Integration Project in Eclipse.
   Go to **File→New→Integration Project**, enter the details of the project and click **Next**.



Select **Point-to-Point Channe**l pattern for the integration project and click **Finish**.



Integration project gets created with below folders.

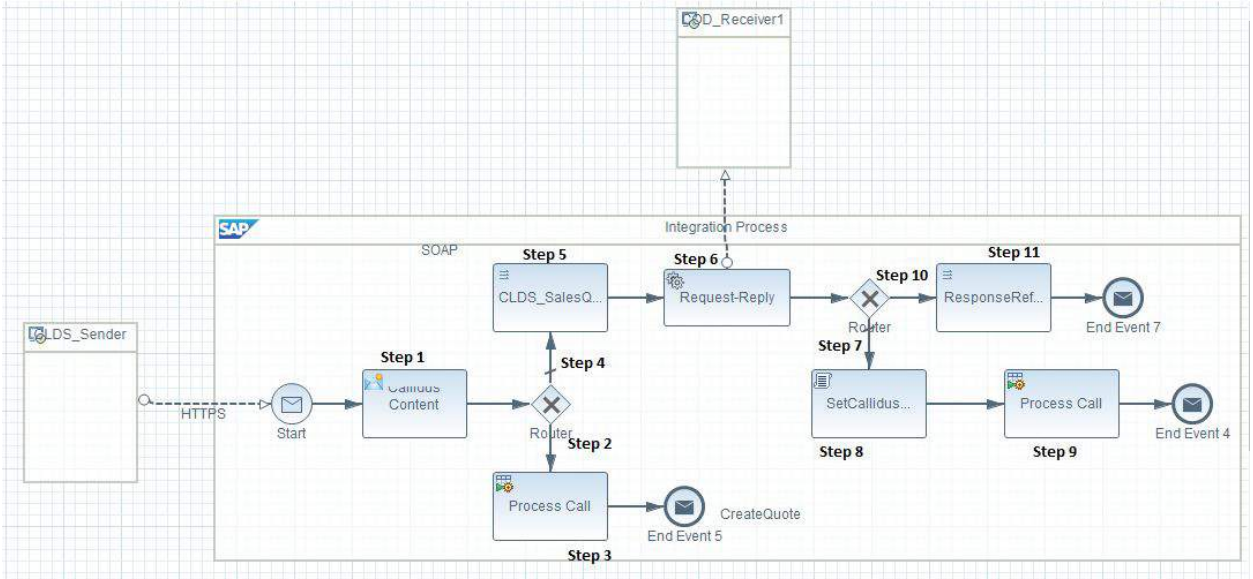


4.4.2 Develop the integration flow diagram

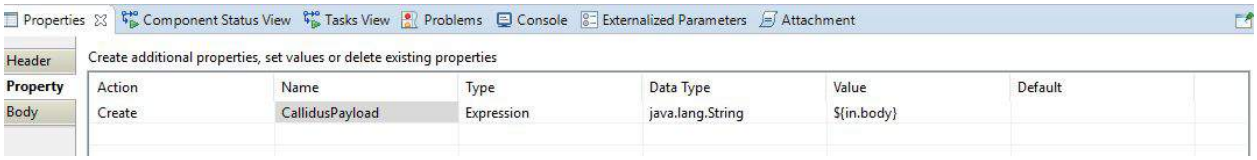
Classification	☒ New ☐ Change of Existing Step ☐ Reuse of Existing Step
Objective	To create a new integration flow for Sales Quote Replication
Involved Users	
Trigger	N/A
Precondition	SCP IS (HCI) developer is provided with the authorization to create an artifact
Post-condition	N/A
End User Interaction with the System	End-user is not involved directly

Open the integration flow file and model it like below.

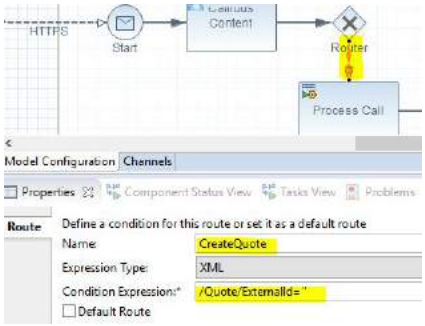
Part 1: Main Integration Process



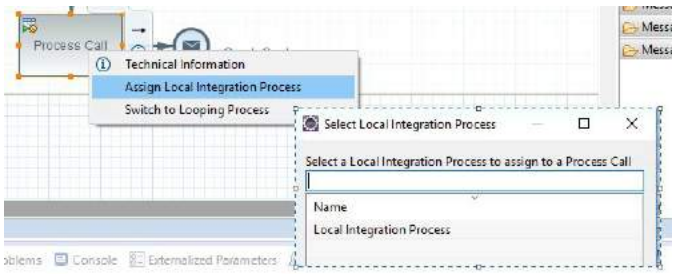
Step 1: Add Content Modifier and create a property "CallidusPayload" to store the incoming Callidus payload



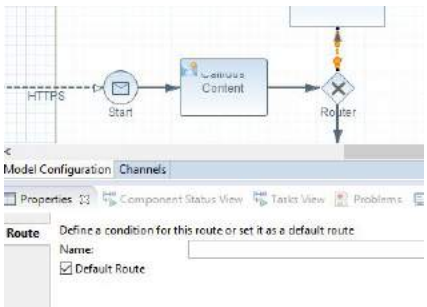
Step 2: Add Router and set a condition expression for the created scenario (when ExternalId is empty)



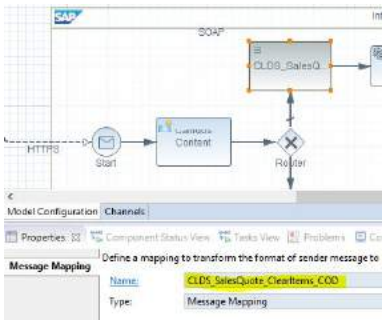
Step 3: Add Process call and assign local integration process (Details are covered in Part 2)



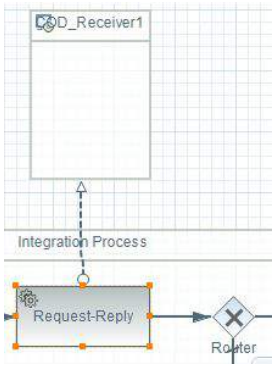
Step 4: Set Route as default (This route is taken for the updated scenario)



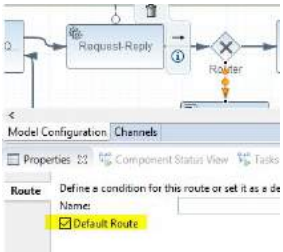
Step 5: Add Mapping component and assign "CLDS_SalesQuote_ClearItems_COD" (Details are covered in message mapping)



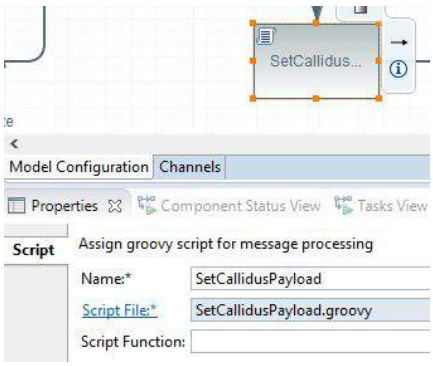
Step 6: Add Request-Reply, Connect to the C4C Receiver and send a response to Router



Step 7: Add a router and Set route as default (when there are no errors in response, this route will be taken)

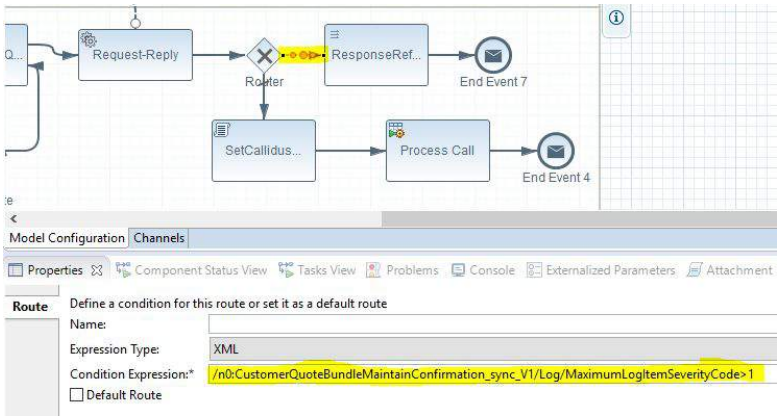


Step 8: Add Script and assign "SetCallidusPayload" (Details are covered in Create script)

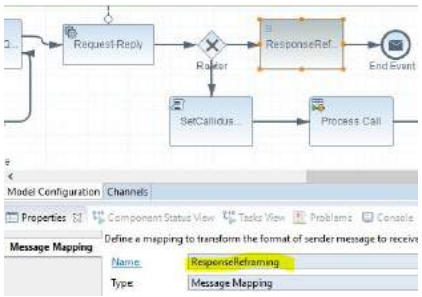


Step 9: The same as Step 3

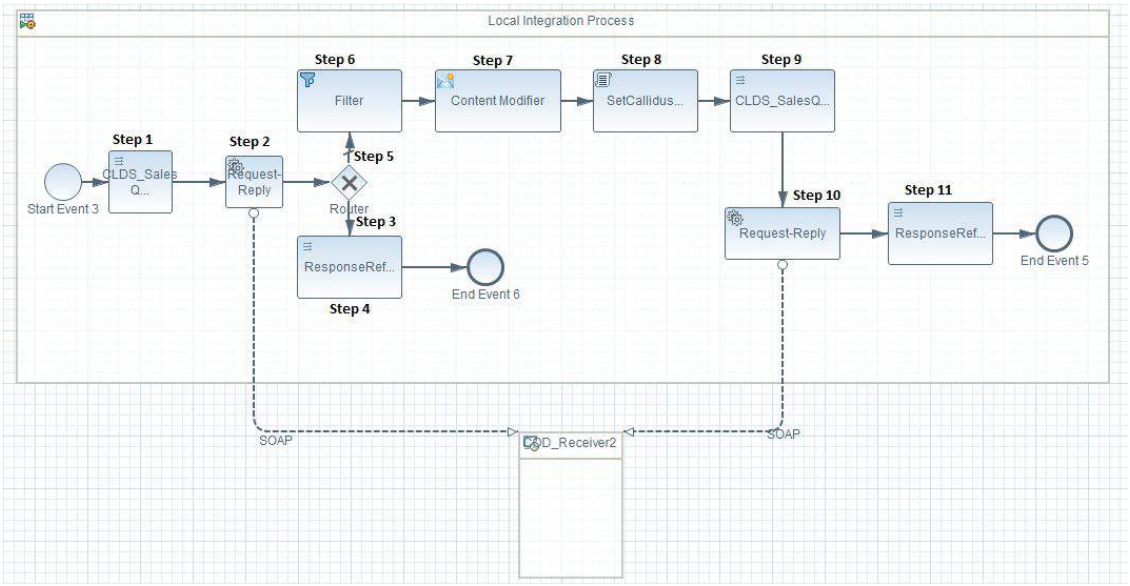
Step 10: Set a condition expression for the route to check for errors



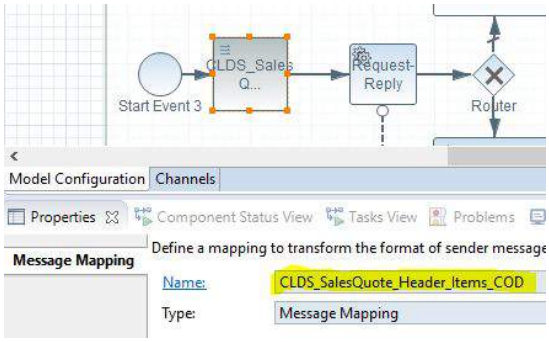
Step 11: Add Mapping and assign "ResponseReframing" (Details are covered in message mapping)



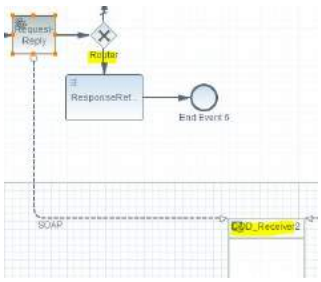
Part 2: Local Integration Process



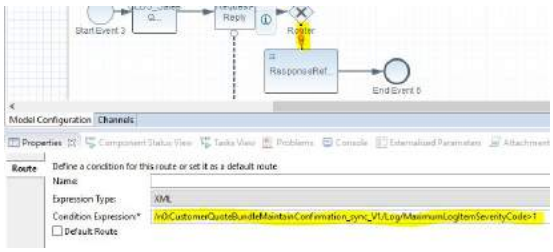
Step 1: Add message mapping and assign "CLDS_SalesQuote_Header_Items_COD" (Details are covered in Message Mapping)



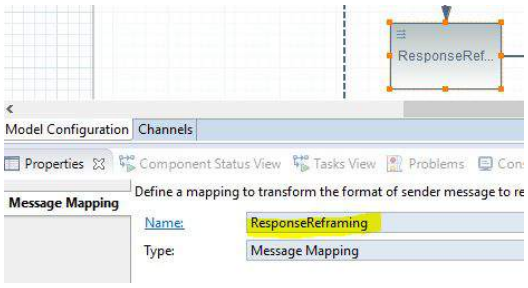
Step 2: Add Request-Reply, connect to Receiver and Router



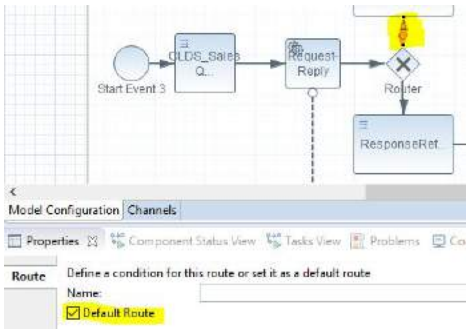
Step 3: Add a Router and set a condition (This route is taken in case of Error)



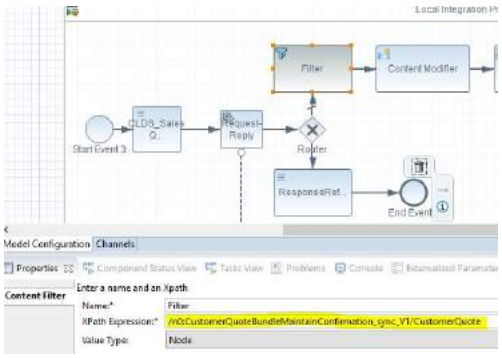
Step 4: Add mapping and assign "ResponseReframing" (Details are covered in Message mapping process)



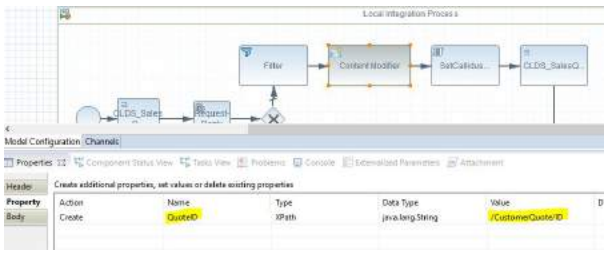
Step 5: Set Route as default (this route is taken when there are no errors to update price component)



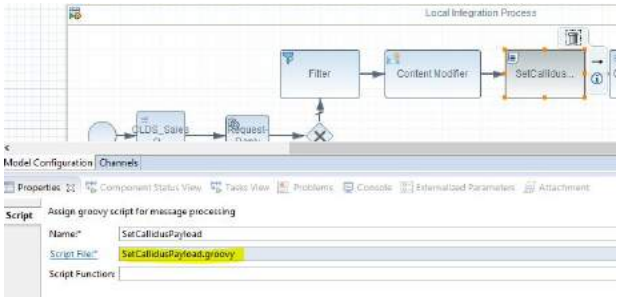
Step 6: Add filter to get node data



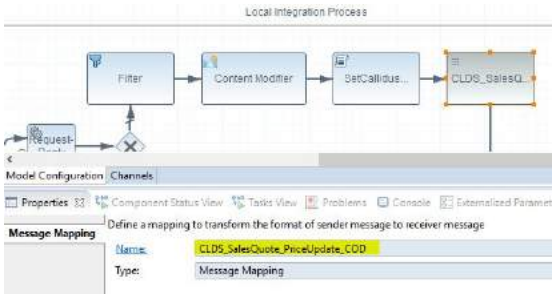
Step 7: Add a content modifier and create property to store C4C QuoteID from the response



Step 8: Add a script and assign "SetCallidusPayload"

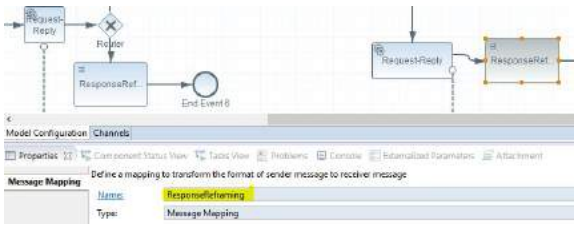


Step 9: Add Message mapping and assign "CLDS_SalesQuote_PriceUpdate_COD"



Step 10: Add Request Reply, Connect to the C4C receiver and ResponseMapping

Step 11: Add Message Mapping and assign "ResponseReframing"



4.4.3 Import WSDLs and schema files

Classification	☒ New ☐ Change of Existing Step ☐ Reuse of Existing Step
Objective	To Import all files which are used for message mapping
Involved Users	
Trigger	N/A
Precondition	SCP IS (HCI) User should be provided with the designer authorization
Post-condition	N/A
End User Interaction with the System	End user is not involved directly

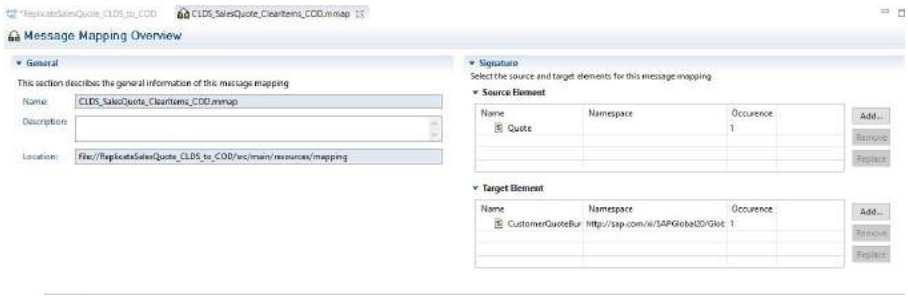
4.4.4 Create message mapping

Classification	☒ New ☐ Change of Existing Step ☐ Reuse of Existing Step
Objective	To map source and target fields for data flow
Involved Users	
Trigger	N/A
Precondition	SCP IS (HCI) User should be provided with designer authorization
Post-condition	N/A
End User Interaction with the System	End user is not involved directly

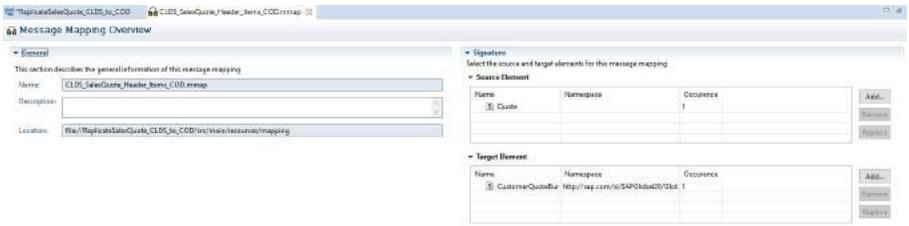
This is done to map source and target fields and four mappings are created in this artefact.

1. **CLDS_SalesQuote_ClearItems_COD** – To clear the line items in case of update sales quote

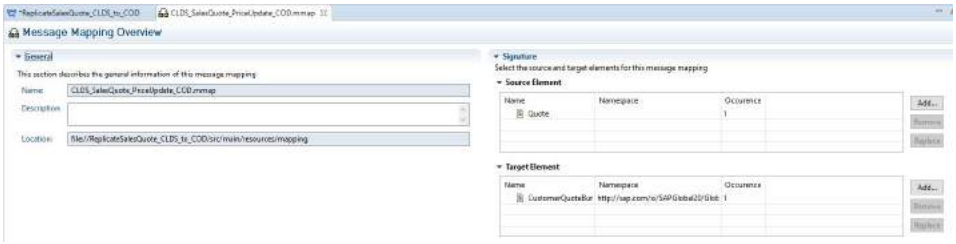
a) Create Message Mapping
b) Import “UpdateSalesQuote” as source and “Manage_Sales_Quote/ CustomerQuoteBundleMaintainRequest_sync_V1” as target



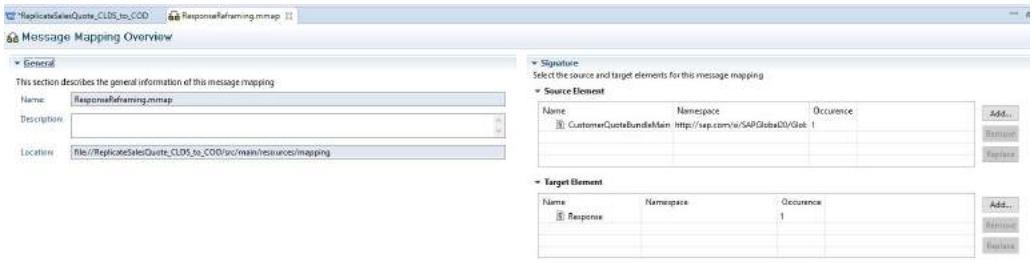
2. **CLDS_SalesQuote_Header_Items_COD** – To Replicate Sales quote header and line items
   a) Create Message Mapping
   b) Import “UpdateSalesQuote” as source and “Manage_Sales_Quote/ CustomerQuoteBundleMaintainRequest_sync_V1” as target



3. **CLDS_SalesQuote_PriceUpdate_COD** – To Replicate Price Components to line items and header
   a) Create Message Mapping
   b) Import “UpdateSalesQuote” as source and “Manage_Sales_Quote/ CustomerQuoteBundleMaintainRequest_sync_V1” as target



4. **ResponseReframing** – To Frame the response in a specific format
   a) Create Message Mapping
   b) Import “Manage_Sales_Quote/CustomerQuoteBundleMainatainConfirmation_sync_V1” as source and “ResponseFormat” as target



4.4.5 Create Scripts

Classification	☒ New ☐ Change of Existing Step ☐ Reuse of Existing Step
Objective	To create scripts using groovy to meet the requirement
Involved Users	
Trigger	N/A
Precondition	SCP IS (HCI) User should be provided with designer authorization
Post-condition	N/A
End User Interaction with the System	End user is not involved directly

A script file “SetCallidusPayload” is created for the following purposes:

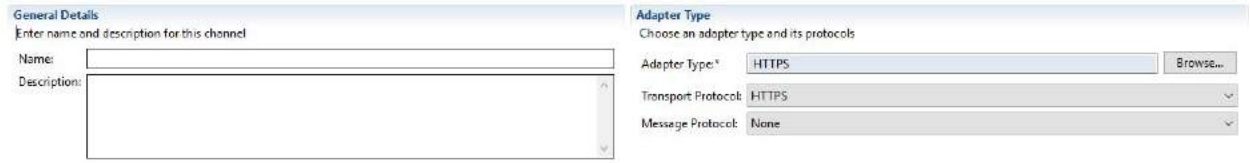
- To fetch the Callidus incoming payload from content modifier Property, to make second call to C4C for updating price components.
- To fetch C4C Quote ID from Response, which is stored in content modifier property.
- To replace empty ‘<ExternalId></ ExternalId>’ tag in Callidus payload with ‘<ExternalId> C4C Quote ID </ ExternalId>’ in Create scenario, as we have to update price components for the correct C4C Quote.

4.4.6 Configure Channels

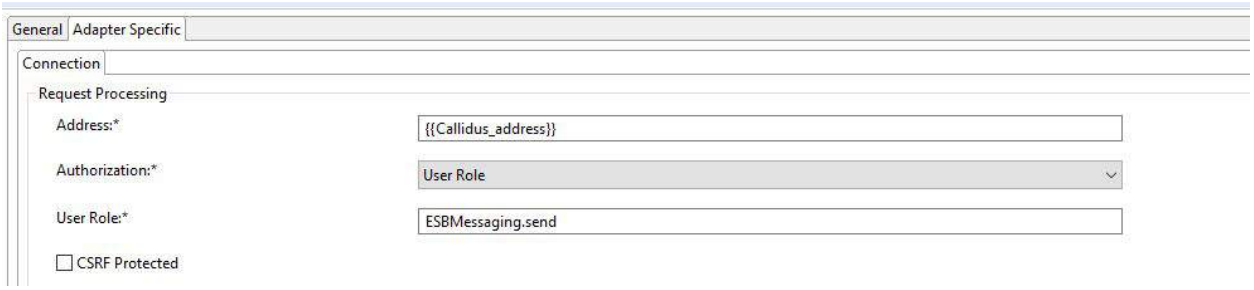
Sender and Receiver channel needs to be configured with below details to receive and send data in the correct format.

<u>Sender Channel</u>

General:

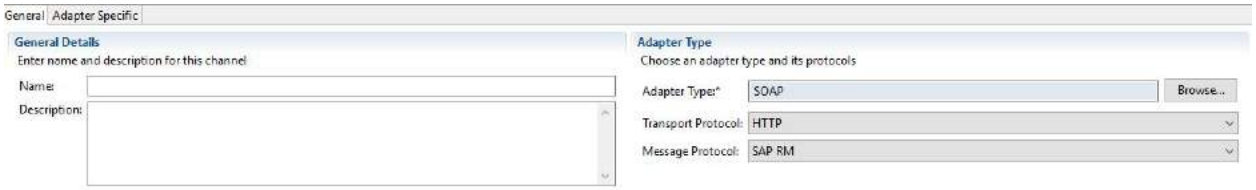


Adapter Specific:

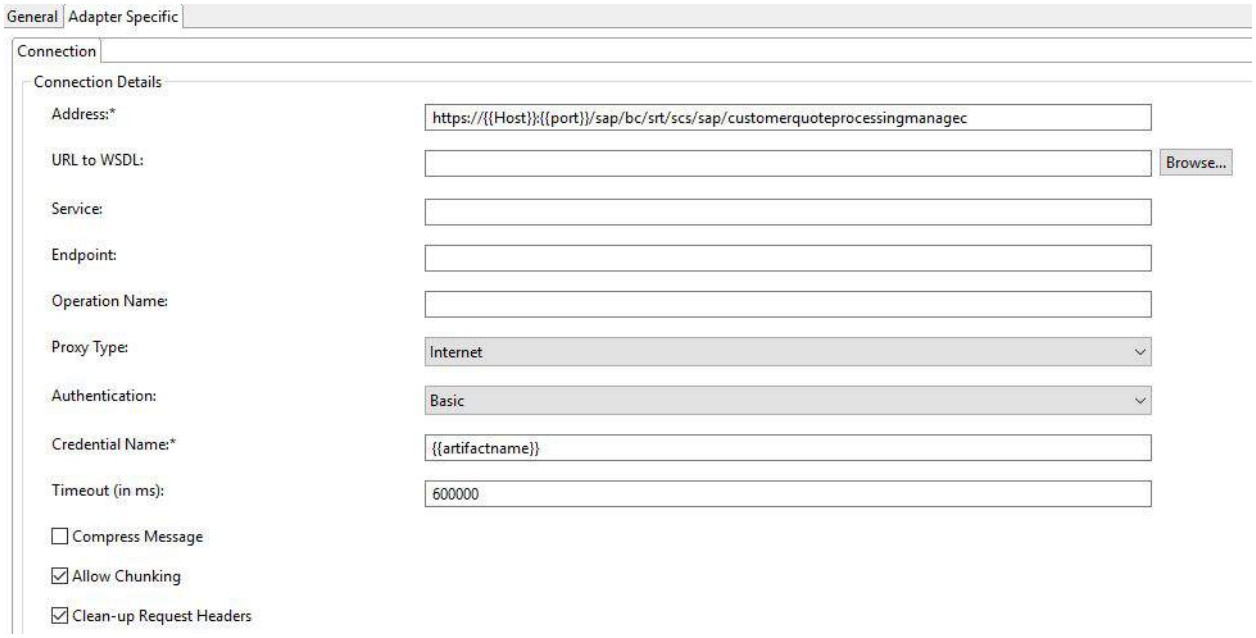


Receiver Channel

General:



Adapter Specific:



Property file: It contains source, target system and credential details.

```
#Mon Jul 24 16:26:52 IST 2017
artifactname=C4C_User
port=443
Host=my319734.crm.ondemand.com
Callidus_address=/CreateUpdateSalesQuote
```

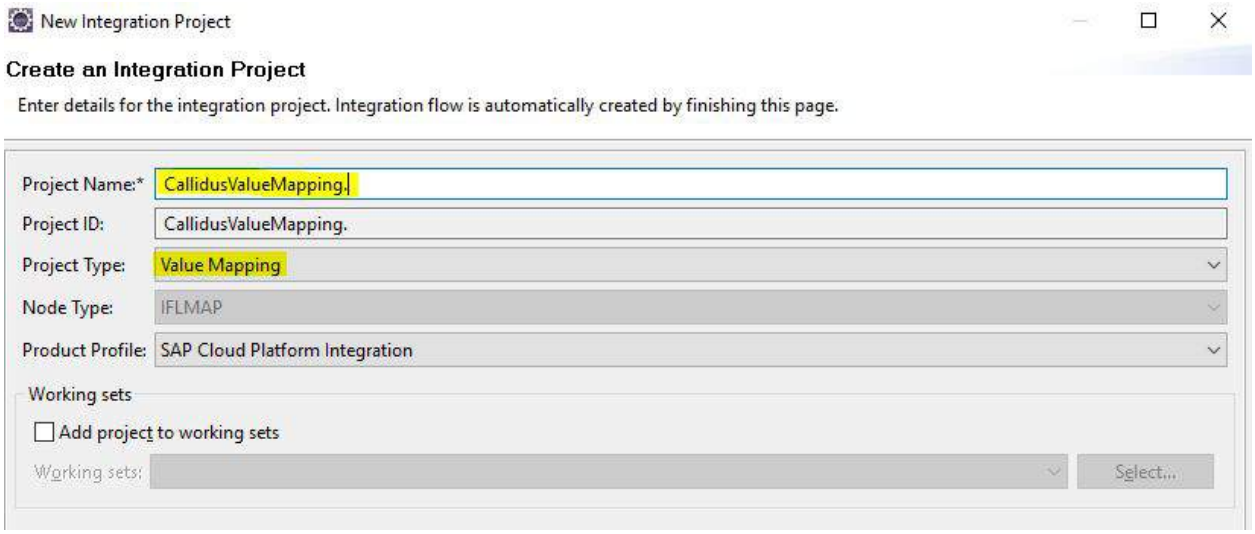
4.4.7 Create Value Mapping

Classification	☒ New ☐ Change of Existing Step ☐ Reuse of Existing Step
Objective	To Create Value mapping used to convert incoming values into SAP C4C Internal Code or Vice versa
Involved Users	
Trigger	N/A
Precondition	SCP IS (HCI) User should be provided with designer authroization
Post-condition	N/A
End User Interaction with the System	End user is not involved directly

Take the following steps to create Value mapping:

1. Go to **File→New→Integration Project**.

2. Enter **Project Name** and Select "**Value Mapping**" in **Project Type.**



3. Open value_mapping.xml and enter the set of values to be considered for conversion.

   In this project, Value mapping is used for:

   a) Price Component Type Code – Callidus System sends only the description which needs to be converted to C4C Internal code.

   b) Severity Code – In SAP C4C Response, Severity code needs to be converted to a description and sent to Callidus.

```xml
<?xml version="1.0" encoding="utf-8"?>
<vm>
<group id="b24bce8c-fc10-42cd-bf1d-040e2b941749" name="ListPrice">
  <entry>
   <agency>CCPQ</agency>
   <schema>PricingTypeCode</schema>
   <value>ListPrice</value>
  </entry>
  <entry>
   <agency>COD</agency>
   <schema>PricingTypeCode</schema>
   <value>7PR1</value>
  </entry>
</group>
<group id="b24bce8c-fc10-42cd-bf1d-040e2b941750" name="ProductDiscount">
  <entry>
   <agency>CCPQ</agency>
   <schema>PricingTypeCode</schema>
   <value>ProductDiscountPercent</value>
   <value>UserDiscountPercent</value>
  </entry>
  <entry>
   <agency>COD</agency>
   <schema>PricingTypeCode</schema>
   <value>7PR6</value>
  </entry>
</group>
<group id="b24bce8c-fc10-42cd-bf1d-040e2b941751" name="ProductDiscountAmount">
  <entry>
   <agency>CCPQ</agency>
   <schema>PricingTypeCode</schema>
   <value>DiscountAmount</value>
  </entry>
  <entry>
   <agency>COD</agency>
   <schema>PricingTypeCode</schema>
   <value>7PRA</value>
  </entry>
</group>
```


Agency is source/target, **Schema** is source/target Identifier and **Value** is what needs to be converted.

Maintain all the values to be considered for value mapping in this XML.

4. Save all the changes and deploy the Integration content.

4.4.8 UpdateOpportunityAndSalesQuote Main IFlow

Classification	☒ New ☐ Change of Existing Step ☐ Reuse of Existing Step
Objective	To Create an Iflow for Update Opportunity and Quote at the same time
Involved Users	
Trigger	N/A
Precondition	SCP IS (HCI) developer is provided with the authorization to create artifact
Post-condition	N/A
End User Interaction with the System	End user is not involved directly

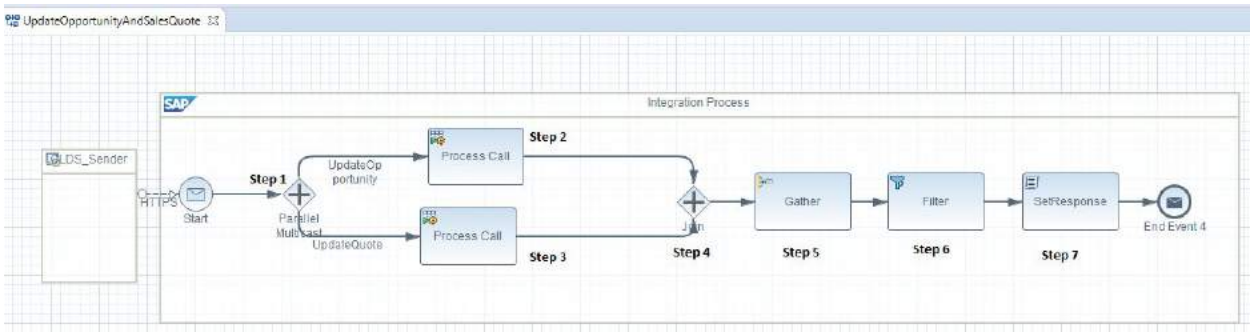
This artifact is created to update opportunity and quote simultaneously.

To create artefact, follow the instructions:

1. Create an integration project.

   Refer to section 6.4.1 for more details.

2. Model the integration flow like in the following example:



   a) Add Parallel Multicast to send the same Callidus payload for update opportunity and update quote.

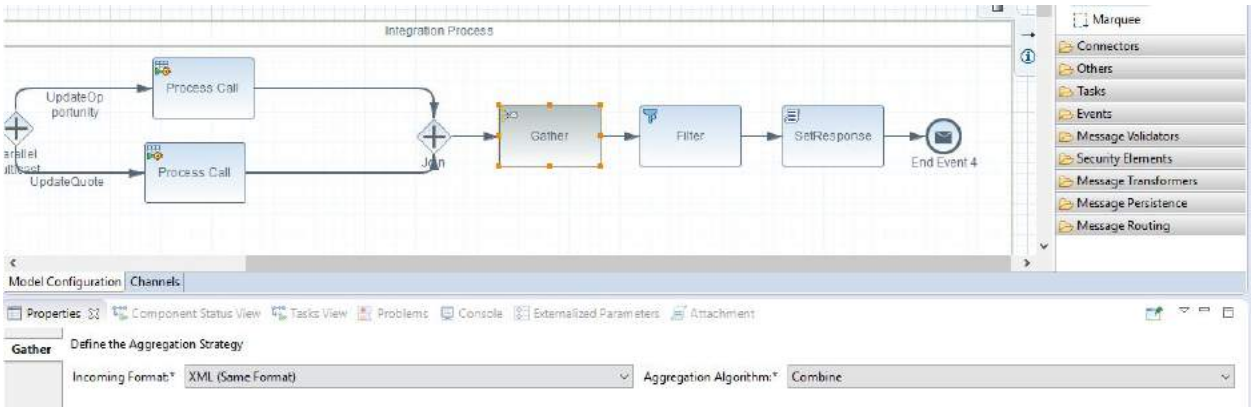
   b) Add process Call and assign Opportunity Integration sub process.

      For more details, refer to 6.3.2 - Creation of Integration Flow Diagram in the technical document for UpdateOpportunity artifact.

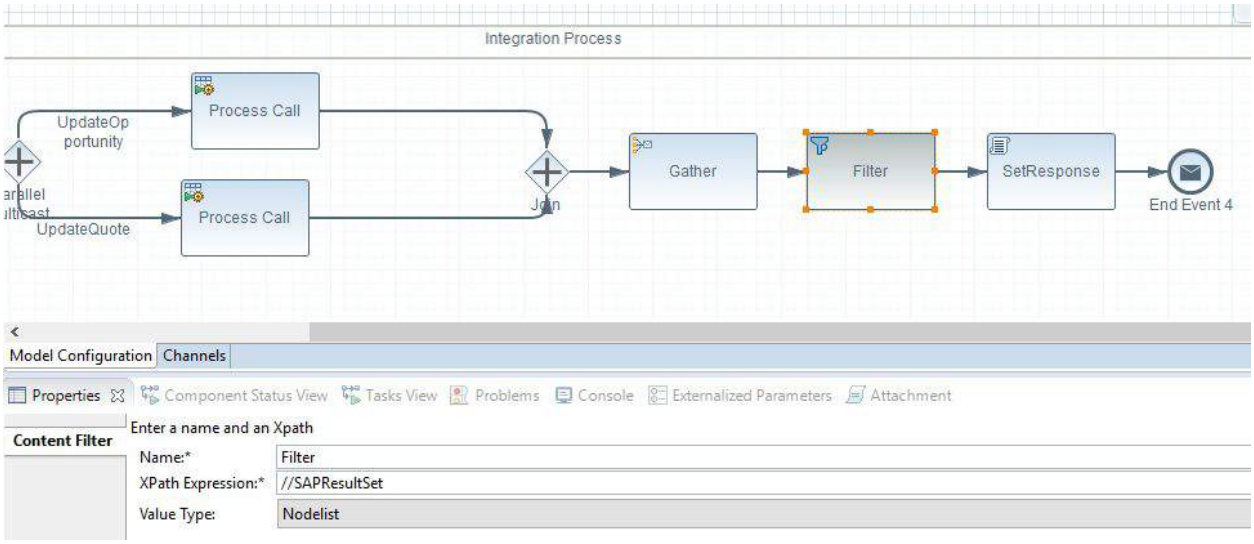
   c) Add another process call and refer to Quote Integration sub process.

      For modelling this, refer to 6.4.2 - DevelopIntegration flow in this document.

d) Join and gather the responses from both the flows.



e) Add Filter to consider only node list of <SAPResultSet> from the combined response.



f) Script is added to frame the Response in correct xml format.

Final XML should look like this:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Response>
    <SAPResultSet>
        <Document>
            <DocumetType>Opportunity</DocumetType>
            <ChangeStateID>                20170801124654.7189540</ChangeStateID>
            <InternalID>46503</InternalID>
            <InfoMessage>Opportunity has been updated successfully</InfoMessage>
        </Document>
        <Success>1</Success>
    </SAPResultSet>
    <SAPResultSet>
        <Document>
            <DocumetType>Sales Quote</DocumetType>
            <ChangeStateID>                20170802072921.7747920</ChangeStateID>
            <InternalID>674</InternalID>
            <ExternalID>POST-1</ExternalID>
            <InfoMessage>Sales Quote Replicated Successfully</InfoMessage>
        </Document>
        <Success>1</Success>
    </SAPResultSet>
</Response>
```


In the script concatenate the filtered node list with proper XML tags.

g) Deploy the artefact.

4.4.9 Deployment of Artifact

Classification	☒ New ☐ Change of Existing Step ☐ Reuse of Existing Step
Objective	To deploy the artifact
Involved Users	
Trigger	N/A
Precondition	Iflow should be saved
Post-condition	N/A
End User Interaction with the System	End user is not involved directly

After the completion of development, the artifact is ready to deploy.